

Constructing Staff Development and Educational Change

Gary S. Stager

Torrance, California
USA

ABSTRACT

Preparing teachers to use technology in educationally meaningful ways is difficult. However, much may be learned from the experiences of schools in which every child has a personal notebook computer. This paper presents staff development models developed for Australian schools with universal computing. The recommendations in this paper are designed for schools interested in creating richer, student centered, learning environments and in preparing students for the future. Practical examples and suggestions for inspiring teachers will be shared.

Main conference themes: teacher education, learner centred learning

Educational areas: primary education

Study topics:

Secondary keywords: Logo, programming

INTRODUCTION

Many educational leaders and policy makers have grand visions of how computer technology will lead to educational innovation and restructuring. Far too many of these people believe that the technology will do the job alone. If staff development is provided at all, it is too often superficial and unsuccessful. Teachers and their students may be 'using computers' but to what end?

The difficult part of the innovation process is not learning the technology, but thinking about thinking and learning, reflecting on the nature of the curricula and clearly articulating a collegiate strategy for implementing change. Computer-based staff development efforts often assume that teachers need to be only computer literate enough to unjam the printer or to use one piece of 'canned software' with their students. This line of reasoning deprives teachers of the types of intellectual empowerment which their students experience when using the computer as a vehicle for constructing knowledge.

Educational reform is too often equated with plugging students into anything which happens to be around to plug in. Although technology is a critical empowering force and intellectual tool, if used properly by schools, it is neither the prerequisite nor vehicle for educational restructuring.

Although the computer-literacy model of teaching children to perform decontextualized mechanical acts with a computer has been discredited, it is ironic that many educational leaders still propose this model for teachers. While there is nothing inherently wrong with using a computer to store grades or write parent letters, this need not be the focus of staff development. I also question the efficacy of making this the teacher's first technological encounter.

Few schools who use such strategies as a 'first step', have a thoughtful plan for connecting the teacher's experience to constructive student use. The way you should benefit teachers is by helping them directly

benefit kids. You improve the lives of teachers by helping them become better teachers. Even the 'bad' teachers our society is so fond of discussing will be inspired by seeing students engaged in exciting new ways—without preset texts and preconceived outcomes. Excitement for learning is contagious.

Therefore computer related staff development should focus on the change process and immerse teachers in meaningful, educationally relevant projects. Such activities should encourage teachers to reflect on powerful ideas and share their educational visions in order to create a culture of learners for their students. In this context it is counterproductive to dispute the universality of every student having a personal computer. This reality for many Australian schools is sure to become the norm in the near future. Schools can either design professional development activities for a computer-rich world or risk the consequences of being caught unprepared.

AUSTRALIAN LEADERSHIP

In 1989, Methodist Ladies' College (MLC) in Melbourne, Australia embarked on a learning adventure still unparalleled throughout the world. At that time the school made a commitment to personal computing, LogoWriter, and constructionism. The governing principle is that every child in the school (from grades 5-12) owns a personal notebook computer on which they may work at school, at home, and across the curriculum. Ownership of the notebook computer is deemed to reinforce ownership of the knowledge constructed with it as it is a vehicle for building something tangible outside of your head—one of the tenets of constructionism. Approximately 2,000 MLC students now have a personal notebook computer.

Personal computing in schools not only challenges the status quo of computers in schools, but creates new and profound opportunities for teachers. Schools often take computers so seriously (which implies hiring special computer teachers and scheduling times at which students may use a computer) that they trivialize their potential as personal objects to think with. Computers are ubiquitous and personal throughout society, just not in schools.

For creating a nurturing learning culture MLC ensured that teachers were supported in their own learning by catering to a wide range of learning styles, experiences and interests. It was agreed that personal computing was a powerful idea more important than the computers themselves. What was done with the computers was of paramount importance. LogoWriter was MLC's primary software of choice (MicroWorlds is now used.)

Dozens of Australian schools are in various stages of following MLC's lead in laptop computing. They follow the MLC professional development model described in this paper.

STAFF DEVELOPMENT

"We start from the assumption that good schools are unique. In order to be good, a school has to reflect its own community. And therefore, we offer no model. There's nothing that you just 'put into place,' nothing to 'implement.' Our research suggests that you're not going to get significant, long-term reform unless you have a subtle but powerful support and collaboration among teachers, students and the families..."

(Ted Sizer, Educational Leadership interview, February 1995.)

While every teacher is expected to use technology in appropriate ways, their learning styles must be respected and catered for via a range of professional learning opportunities. Teachers at 'laptop schools' have stated that sharing ideas with colleagues and residential events have been their most rewarding staff development experiences.

Teachers are introduced to computers by being challenged to reflect on their own learning while solving problems of personal significance in the software environment, LogoWriter—the software the students would be using. I would argue that educational progress occurs when a teacher is able to see how the particular innovation benefits a group of learners. These teachers come to respect the learning processes of their students by experiencing the same sort of challenges and joy. The teacher and learner in such a culture are often one-and-the-same. Other teachers find the enthusiasm and pride of their colleagues infectious. Schools use LogoWriter to help free the learner to express herself in unlimited ways—not bound by the limits of the curriculum or artificial (school) boundaries between subject areas.

Staff Development Innovations

Many schools find the task of getting a handful of teachers to use computers at even a superficial level daunting. In contrast the 'laptop schools' expect their teachers to not only be comfortable with thirty notebook computers in their classroom, but to participate actively in the reinvention of their school. In such progressive schools staff development is no longer a technical act of pouring information into a teacher's head or training them in a few technical skills.

Staff development should help teachers fearlessly dream, explore and invent new educational experiences for their students. Staff development experiences must embody powerful ideas and provide teachers with the opportunity to discover the learner inside and to fall in love with learning. Such a teacher will also love to teach. Modern staff development must help teachers embrace not only the technology, but anticipate the classroom change which will accompany widespread use.

I have employed three staff development strategies: in-classroom collaboration, slumber parties and build-a-book workshops in several 'laptop schools'. All model constructionism by providing meaningful contexts for learning, by an emphasis on collaborative problem solving and personal expression, and by placing the learner (in this case teachers) at the centre. Each strategy values and respects the professionalism of the teacher by acknowledging the knowledge, skills and experience they each possess. Teachers hopefully recognize the respect afforded them so that also the idea that 'kids know stuff' will become a basic element of their teaching.

In-classroom collaboration

Several Australian 'laptop schools' have used the in-classroom model I developed working in the Scarsdale, NY and Wayne, NJ public schools. This collaborative form of teacher development places the 'trainer' in the teacher's classroom to observe, evaluate, answer questions and model imaginative ways in which the technology may be used by the learners. The collaborative spirit and enthusiasm engendered by the 'trainer' motivates the classroom teacher who feels more comfortable taking risks when a colleague is there to help. Since this professional development occurs on the teacher's turf and during school hours, it makes implementation more viable.

Residential Logo 'slumber parties'

Teachers leave the pressures of school and home behind for three days to improve their computing skills in a carefully constructed environment designed to foster opportunities for peer collaboration, self expression, personal reflection and a renewed enthusiasm for learning. These learner centred 'workshops' stress action and collaboration, not rhetoric. The workshop leader serves as a catalyst and creates opportunities for connecting personal reflections to teaching. These connections are powerful when they come from the teacher's own experience—much like the types of learning opportunities we desire for students.

The constructionist slumber parties consist of three key elements:

- **Project brainstorming:** Before we are even sure that every teacher can turn on the computer we ask them to identify LogoWriter projects they wish to work on for the next three days. The projects may be collaborative or personal—curriculum-related or having nothing to do with the subject they teach. This brainstorming session assures teachers that they will be able to change their minds, but stresses that they must 'get busy.'
- **Powerful ideas:** Each day begins with a discussion of a related education issue and often a philosophical discussion. Such topics might include: 'The history of Logo and your role in technological innovation' (what the school has already accomplished), process approaches to learning or sharing a personally profound learning story. The final day's topic, 'What does this have to do with school?' seeks to have teachers reflect on their recent learning experiences and make connections to their role as teachers.
- **Problem solving off the deep end:** One or two problem solving activities of a two or three hour duration are organized to demonstrate how complex open-ended problems may be solved via collaboration and the sharing of expertise. These experiences often result in an awareness that not every problem has one correct answer or is even solvable. The teachers gain an appreciation for how much can be learned through such a problem solving process and how these activities usually require more than 45 minutes of class time.

Such 'slumber parties' are organized on a regular basis in an attempt to enlarge the pool of teachers comfortable with similar learning philosophies. Since the primary goal of the workshop is to support a learning community, teachers and administrators are encouraged to participate in these workshops more than once. Participants also gain appreciation for the power and expressive potential of LogoWriter. They are reminded that their colleagues are creative, imaginative learners like themselves.

Build a book programming residential workshops

The teachers found that kids could do a lot with a little Logo, but expressed a need to learn more formal programming techniques. I was concerned with finding a format which would enhance the teachers' programming fluency without lecturing and traditional instructionist computer science pedagogy.

My solution to this challenge was inspired by Chip Healy's book, *Build a Book Geometry* [1]. The book chronicles Chip's experience as a high school geometry teacher who spends the entire year encouraging his students to write their own geometry text through discovery, discussion, debate and

experimentation. The book provides an exciting model for starting from what teams of students know and can articulate about a concept, and then giving them challenges building upon their understanding or misunderstanding. The responses are then used to elicit a set of issues to be responded to by another team and so on. Throughout this process each team is encouraged to keep careful notes of hypotheses, process and conclusions, and to share these notes with the other teams during the process of writing the class book. I sensed that this wonderful approach to constructionism could be used to 'teach' complex programming issues in two days.

Healy's ideas inspired a format which addressed the areas of confusion and needs of each teacher through collaboration, discussion, problem solving and journal writing. Before the workshop I asked each participant to identify three LogoWriter programming issues which they did not understand or needed help with.

Small teams of teachers spend hours answering the questions and explaining numerous programming (and often mathematical issues) to each other. This exercise stresses the most important component of cooperative learning, interdependence.

When each group had answered all four questions to their collective satisfaction, each teacher met with a member of another team and explained what each team had accomplished. This peer teaching takes a substantial amount of time.

Emerging questions are further explored through projects designed by the workshop leader and intended to utilize increasingly sophisticated skills. Issues of programming elegance are dealt with by simplifying strategies commonly observed in student projects. Seemingly complex issues associated with multiple turtles, recursion, list processing global versus local procedures, data representation and inputs become clear.

The teams are asked to keep careful notes of their programming processes, questions and discoveries—in addition to answers to the problems. These collective notes are included in the class 'book' (disk). This disk is viewed as a valuable 'personal' reference source the teachers can use back in their classrooms.

Teachers' own assessment of the residential workshops have been extremely positive. The quality of the experience makes the cost of sending fifteen teachers to the Hilton for three days inexpensive when compared with the cost of a never-ending series of two-hour after-school workshops. Schools routinely spend a lot more time teaching concepts in bite-size chunks, while leaving real learning to chance.

Suggestions for Success

Stay on message: School administrators need to articulate (and believe in) a very clear philosophy regarding how the new technology is to be used and how the culture of the school is likely to change. Communication between teachers and their administrators has to be honest, safe and comfortable. Administrators need to constantly clarify the curricular content and traditions the school values, as well as the outdated methodology and content which may be eliminated. Teachers need to trust their administrators to support them through these transitional periods.

Work on teachers' turf: Those responsible for staff development should be skilled in classroom implementation and work along-side the teacher in her classroom to create models of constructive

computer use. It is important for teachers to see what students are capable of, and this is difficult to do in brief workshop at the end of a long work day.

Work with the living: Schools have limited technological and teacher development resources, and these should be allocated prudently. Good teachers who have yet to recognize how computer technology may enhance their teaching, are not evil. If a school focuses its energy and resources on creating a few successful models of classroom computing each year, the enthusiasm among teachers will be infectious. A selection of models must be offered to teachers of differing backgrounds and subject areas. The school should be cautious not to create negative models of computing use.

Off-site Institutes: Schools must ensure that teachers not only understand the concepts of collaborative problem solving, cooperative learning and constructionism—they must also be given the opportunity to leave behind the pressures of family and school for several days in order to actually re-experience the art of learning with their colleagues. Off-site residential 'whole learning' workshops can have a profoundly positive effect on a large number of teachers in a short period of time.

Provide adequate resources: Nothing dooms the use of technology in the classroom quicker than a lack of support for the teacher who worked hard to develop new skills. Be sure that the school does everything humanly possible to support the teacher's efforts by providing the technology requested, maintaining it and by providing access to a working printer and a supply of blank disks. Scarcity is implementation's greatest obstacle.

Avoid software "du jour": The people responsible for paying for school computing are made to feel guilty by the media and other administrators if they do not constantly do something 'new' with their computers. Unfortunately newness is equated with lots of software. It is reckless and expensive to jump on every software bandwagon. Using narrow skill specific software has little benefit to students and undermines staff comfort with computing. Choose an open ended environment, such as MicroWorlds, in which students express themselves in many ways that may also converge with the curriculum.

Practice what you preach: Staff development experiences should be engaging, interdisciplinary, collaborative, heterogeneous and models of constructionist learning.

Share learning stories: Teachers should be encouraged to reflect on significant personal learning experiences from their lives and the staff development experience. They should share these experiences with their colleagues and discuss the relationship between their own learning and their classroom practices.

Celebrate initiative: Teachers who have made a demonstrative commitment to educational computing should be recognized by being freed of some duties in order to assist colleagues in their classrooms, encouraged to lead workshops and given access to additional hardware.

In-school sabbaticals: Innovative teachers should be provided with the school time and resources necessary to develop curricula and conduct action research in her/his school.

Assist teacher purchases of technology: Schools should help fund 50-80% of a teacher's purchase of a personal computer for use in school and home. This act demonstrates to teachers a shared

commitment to educational progress. Partial funding gives teachers the flexibility to purchase the right personal computer configuration. The school may offer an annual stipend for upgrades and peripherals.

Cast a wide net: No one method of staff development works for all teachers. A combination of traditional workshops, in-classroom collaborations, mentoring, conference participation and whole learning residential workshops must be available for teachers to choose from at their own pace. Therefore, a variety of staff development options may need to be offered regularly.

Although most schools can dream of only a handful of computers, the reality of what is happening in schools all over Australia requires serious consideration. Professional teacher development must accept the eventuality of universal computing in order for schools to have a relevant future. If we desire to restructure schools, then we must recognize that the only constant we can depend on is teachers. Our schools will only be as good as the least professional teacher. Staff development must enhance that professionalism and empower teachers to improve the lives of their students. Our children deserve no less.

REFERENCE

1. Healy, C. (1993). *Build-A-Book Geometry*. Berkeley, CA: Key Curriculum Press.

Stager, G. S. (1995). Constructing staff development and educational change. In D. Tinsley & T. J. Van Weert (Eds.), *World Conference on Computers in Education VI: WCCE'95 Liberating the Learner, Proceedings of the sixth IFIP World Conference on Computers in Education, 1995* (pp. 1079–1087). Birmingham, U.K.: Chapman & Hall on behalf of the International Federation for Information Processing (IFIP).
