Why 2009 Was a Transformational Year for Education

David Thornburg, PhD
Thornburg Center
www.tcpd.org
dthornburg@aol.com

Long after the steamed arguments surrounding No Child Left Behind have faded into the remote mists of history, 2009 will be remembered by educational historians as the year when schools that did not provide laptop computers to every child paid a premium to do so.

One could have argued for years (and many did) that personal computers in the hands of all children afforded an opportunity for an improved educational experience. Those few schools with robust one-to-one programs showed positive results, especially when computers were used as open-ended tools, not just as electronic textbooks. Such benefits aside, many districts said they simply could not afford to provide computers for every child, no matter how beneficial they might be. But, by 2009, the cost argument disappeared. In fact, that was the year that the price of powerful computers dropped significantly below the cost of paper textbooks. After 2009, some might argue that failure to provide laptops to every child might constitute institutionalized child abuse.

Let's do the math. A single school book costs a district about $75 or more. With each student having about four textbooks, the total cost for this outdated technology is likely to exceed

$280. For only $230 or less, each student could be provided with a powerful netbook computer filled with high quality software. This is a savings of $50 per student. For a school with 500 students, the total savings would be $25,000 – money that could be spent on increased staff development, and other infrastructure investments. The savings for large districts would climb into the multi-millions – all in support of providing improved educational experiences for children.

As for the netbook computers available in 2009, I did an experiment. Using one of these inexpensive computers I wrote, typeset, and created printer masters for a new book that was published by the end of the year. This task included editing images, crafting page layouts, etc. In other words, I was performing tasks beyond those expected of most students in K-12 schools, and the netbook was more than adequate for the task.

Other benefits include the fact that a netbook tips the scales at about one kilogram – a fraction of the weight of a book bag filled with paper textbooks.

Of course those who wish to re-enact King Canute and try to stop the tide will argue that many schools lack enough electrical capacity to charge all these computers. Well, the one I bought has a six-cell battery that lasts for six to eight hours of continuous use. That is an entire school day. If the students bring their computers from home fully charged, they can work the entire day without having to recharge the batteries. The battery charger draws about 30 watts – about as much as the light bulb inside your refrigerator. From the school's perspective (again using 500 students), the power demand for the school is 500 students times thirty watts times zero (the computers get charged at home) – which works out to zero additional watts!

Now the cautious reader will note that I said these computers would be loaded with great software, all within the original hardware cost. They key to this cost saving is to use Free and Open Source Software (FOSS). This software operates on the premise of other open source-endeavors – give a brick, get a house. Every person using this software is encouraged to become an informal part of the development community, even if one's contributions are limited to reporting bugs as they are found. The result is software that gets better with age, often at a pace far more rapid that that of proprietary software.

Let's start with the operating system.

Now, to be honest, I've long championed the idea that the quality of software used by students should be at least as good as the quality demanded by professionals using the same

kinds of products. Watered-down educational software never made sense to me, and still doesn't. As it turns out, this is not a problem. There are FOSS programs for a large number of educational applications, all of which have high quality and all of which have rapid bug repairs when defects are exposed. Rather than get into the food fights surrounding the tired Windows/Mac debate, let's cut to the chase and install Linux. This operating system has been around for a long time, and is built on the thinking behind UNIX, an operating system used continuously in industry since the 1970's. The Linux core is overlaid with a graphical user interface, most popularly, Gnome. Computers using this operating system have desktops virtually indistinguishable from the proprietary interfaces known to most computer users. By 2009, the process of installing, setting up, and maintaining software under Linux had become so automated and simple that it is now easier to put Linux on a new computer than it is to install any version of Windows.

Let me give an example. I took a laptop that was two years old and did a clean install of the Linux version called Ubuntu 9.10 (www.ubuntu.com). During the setup, I was asked a series of questions as the installer checked out my hardware. For example, it detected that my Nvidia graphics chip had new drivers available, and asked if I wanted them. (I said yes). It did the same with my internal Wi-Fi card, webcam, USB microphone, etc. Once the operating system was installed and the system rebooted, it then starting looking at my local network where it found my color laser printer. Again, it asked if I wanted to set up the printer, and I said "sure." within a short period of time, my system was not only ready to use, it had found my peripherals, installed a plethora of applications, and even checked on-line to see if any of the applications had been updated recently.

Now, to be honest, this ease of setup was not always present in Linux. It was common for me to have to use the "terminal" in which arcane commands were typed to do simple tasks like install new typefaces, etc. Now that process has become fully intuitive (e.g., double click on a typeface icon. If it isn't already installed, you can do this with a single click of the mouse.)

As for compatibility, Ubuntu 9.10 recognizes any hard drive you connect externally – Windows and Macintosh included! This makes file transfer from any other system trivially easy – and is a feature not found in any proprietary operating system I know of.

As for the software itself, amazingly high quality applications abound – and many of these work on other operating systems as well so a student can work on the same project using a variety of computers, all running the same free program.

One of the main programs found on most computers is an "office suite," a series of programs

for the creation of various kinds of documents. In the open source world, OpenOffice.org is the dominant player. Prior to 2009, OpenOffice.org had some features that were better than proprietary juggernauts like Microsoft Office, and some that were not as good. By late 2009, OpenOffice.org had reached the point where it either mets or exceeded the capacity of Microsoft Office – and it doesn't cost a dime.

It is hard to overstate the importance of having quality software available for free – not only does this save schools a tremendous amount of money in licensing fees, it means that students can use the same software on a home computer they are using at school. Teachers are free to hand out disks filled with software installers that kids can take home with them. Michigan City Area Schools, in Michigan City, Indiana, has gone a step further and built a "Freedom Toaster" in the lobby where students can insert blank disks and download complete software collections in a fraction of the time needed to download some of these titles on-line.

The list of powerful FOSS applications germane to education is huge and growing rapidly. Programming languages like *Scratch* (from MIT), graphics editing tools like *GIMP*, math software like *Geogebra*, and myriad other titles devoted to everything from science to web authoring to movie making are all available for free.

This doesn't mean that there aren't some excellent proprietary programs in education worth buying – there are plenty of them. But, with the money saved by using FOSS for the main applications, there is plenty of money left for these special titles – more, in fact, than is currently available for schools locked into licenses for proprietary operating systems.

All these benefits aside, there are still some who remain convinced that a textbook-driven curriculum meets the needs and learning styles of today's student. Without going into this argument, suffice it to say that there is a no-cost solution to this problem as well: Open Source Textbooks (e.g., www.ck12.org). Under an initiative sponsored by the Governor, California has taken the leadership in exploring this option, and several open source textbooks have already been approved for adoption by the state. Using a concept called "FlexBooks," this site allows teacher to craft their own versions of textbooks from a growing library of modules. The final "book" is created as a PDF file that can be downloaded to a student's computer. If this is too much effort, complete compilations of books, based on state standards, are also available for the effort of a mouse click. Because the material is released through a Creative Commons copyright, users are free to share, modify and otherwise use the books as they see fit.

The first time I saw this approach I thought of my high-school US History teacher. His PhD thesis was on the politics surrounding the construction of the Panama Canal and he would regale us with stories of the events taking place below the radar during that huge project. We not only learned history, we learned that being a historian might be very cool! Were he teaching today, he could upload his own chapters on the topic to CK12, and then assemble a customized textbook for us to use. Furthermore, other history teachers could access his insights and (should they wish) incorporate them into their own custom books.

Detractors may feel that students expect bound paper textbooks, not electronic files. Aside from the fact that a printed file will still cost less than a commercial textbook, I think some teachers are reflecting their bias for information in printed form. I've heard kids ask for many things – a new computer, a Nintendo Wii, a Playstation 3 – but never have I heard a kid say: "What I really want for Christmas is a 600 dpi wireless color laser printer." Think about this. Many computer retailers toss in a printer for free when you buy a computer. They must have warehouses filled with these things! Today's kids are not called "screenagers" for nothing.

That said, I still hold that there are many better ways for children to learn than through the teacher-mediated textbook – ways that take advantage of modern personal computers in the context of modern pedagogical models, such as the constructionist model of Seymour Papert.

But how do I make the change?

Good question. You already have a warehouse filled with textbooks, and a whole bunch of computers in your schools for which you may be paying a small fortune in software licenses.

Here's what I'd suggest. First, as software licenses for Office come up for renewal, let them lapse and move to OpenOffice.org. Because this software exists for Windows, Macintosh and Linux, it will run on any computers capable of running Office. This can save thousands of dollars without losing any quality or capability in the software you use. In fact, I make this offer: If there is an important task you want to do that OpenOffice.org (latest version) can't do, send me an e-mail. If I can't show you how to do the task, I will send you a free copy of one of my recent books. (Now, before contacting me, be sure to scan the on-line help for OpenOffice.org). As you start moving to more and more FOSS applications, you will have the financial resources you need for rich materials such as those offered by Discovery, LVC, NetTrekker, Tech4Learning and others.

Next, as your textbook cycle moves to the next adoption phase, spend the money on powerful netbooks for the children instead. This way you will be phasing in universal computer access to every child and saving even more money in the long run. By the way, as classrooms move

from away from textbooks, be sure that all teachers are supported with ongoing staff development. The importance of investing in teachers can not be overstated. In the event that textbooks remain essential for some teachers, look for the materials you need at sites like CK-12 where rich and well-written textbooks can be downloaded for free.

Yes, the move may feel strange at first, but experience in large scale adoptions has shown that the move is well worth the effort.

So what made 2009 important in education? It was simply the first time that schools had to pay a premium for keeping the powerful learning tool of computers *out* of their students' hands. How often is it the case that doing the right thing is also the way to save money?

The time for excuses is over.

About the author:
David Thornburg is the Founder and Director of Global Operations for the Thornburg Center for Space Exploration. He is an award-winning futurist, author and consultant whose clients range across the public and private sector throughout the planet.

David is a strong proponent of computer use by students, and is a long-time advocate of Linux and other open source programs in education. His presentations and workshops on this topic have been given throughout the world.

His educational philosophy is based on the idea that students learn best when they are constructors of their own knowledge. He also believes that students who are taught in ways that honor their learning styles and dominant intelligences retain the native engagement with learning with which they entered school. A central theme of his work is that we must prepare students for their future, not for our past.

About the Center:
The Thornburg Center has, among its members, Mike Huffman, a recognized pioneer in large-scale implementations of Linux and FOSS in education. Both he and Dr. Thornburg are available to conduct workshops and help plan and deploy FOSS in settings of any size. Visit www.tcpd.org for information on David and Mike, and contact the Center through dthornburg@aol.com to arrange a workshop or consultation visit to your site.