
THE MISSING REVOLUTION

AMD and Novell's Role in bringing Powerful Computers to Every Child

David D. Thornburg, PhD
Director, Global Operations
Thornburg Center
dthornburg@aol.com
847-277-7691

Executive Briefing: 602A
Thornburg Center
711 Beacon Dr.
Lake Barrington, IL 60010
www.tcpd.org
www.tcpdodcast.org



*This document may be freely copied, posted, and distributed in its entirety.
Contact the Thornburg Center for other uses.*

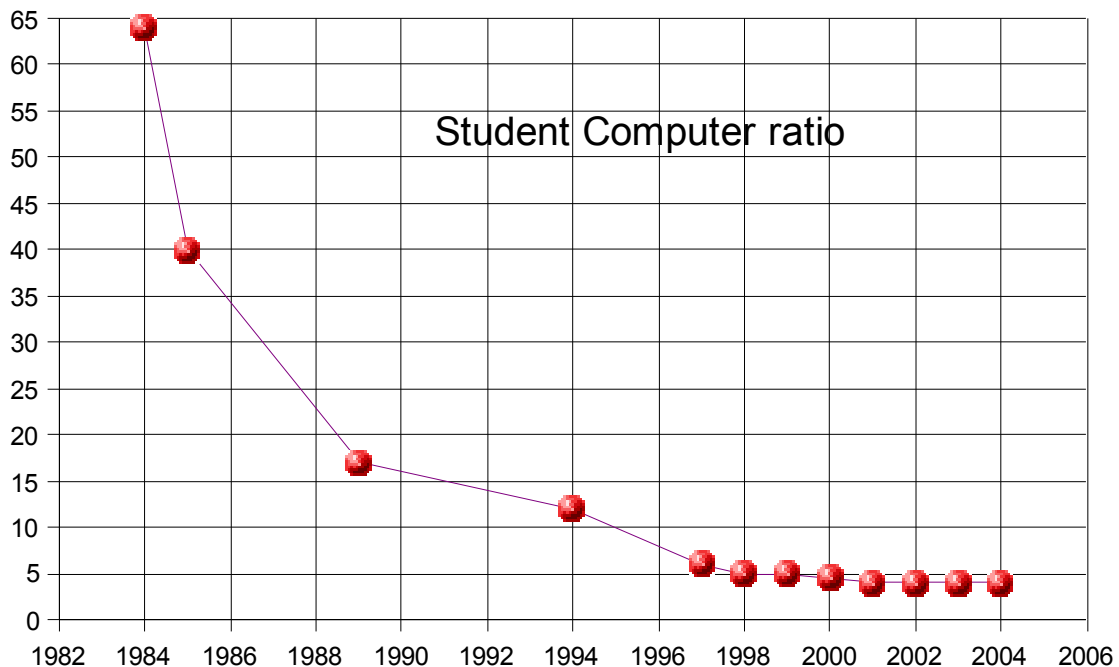
In This Whitepaper

Because of the revolutionary promise it holds in teaching and learning, achieving a level of one computer per student has become the goal in educational computing. While great progress has been made in getting more computers into schools, computer penetration on a per-student basis has long been stalled at only 25%. This whitepaper explores obstacles that have contributed to this stall and tangible paths for overcoming them with technology that is available today.

Computer Penetration In Schools

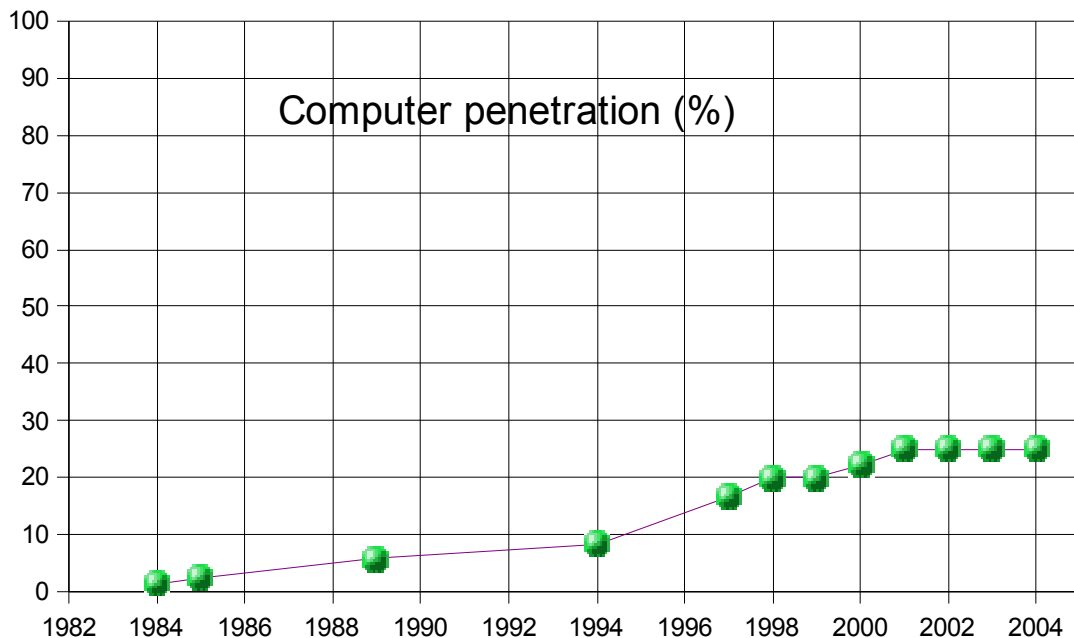
I first entered the world of microprocessor-based personal computing for education in the late 1970's. In those days, the dominant players were Commodore, Radio Shack, and Apple. Computer software was often written by teachers themselves and either given away, or sold for reproduction costs on tape cassettes. Over the following years, educational computing came to be increasingly popular, and the number of computers soon reached the level of one per school, with the next goal being one computer per classroom. Today, with sharply reduced prices for hardware, we talk about one computer per student as a goal.

One-to-one computing promises to promote a revolution in teaching and learning, but while the two goals of bringing computers to schools and then classrooms were achieved fairly quickly, the move to one-to-one computer access has been much harder to achieve. Even so, there are those who claim we are on the path to universal access, and use charts like the following to show how much progress has been made in bringing computers into schools.



The above graph of student-computer ratio (based on data from the National Center for Educational Statistics) is impressive at first glance. From a ratio of sixty-five students per computer as recently as 1984, the curve shows a steep decline to the present ratio of four students per computer. The challenge presented by this data is not apparent when plotted in this fashion, and one can be forgiven for thinking that we are continuing to make progress toward universal access when, in fact, this is not even remotely the case in U.S. schools.

To see why, let's plot the same data a bit differently. Instead of plotting student-computer ratio, let's invert the dataset and look at the percentage of students who (at any given moment) have access to technology in their schools.



The above graph tells a compelling story. Over the past four years, computer penetration in U.S. classrooms has leveled off at 25% (on a per-student basis), and shows no sign of changing. This means, if Juan is using the computer, then Maria, George and Phyllis are not. With penetration this low, how could we possibly think that computers have a transformative impact on education?

Schools are, in fact, buying computers every year. Where are they going? New technology purchases by schools have largely been devoted to replacing outdated or broken equipment, leaving 75% of the goal unachieved. Budget cuts at all levels of government have contributed to the problem, making it seem that we will never be able to fill in the gap. As we'll see later, there

are some steps that can be taken to help, however, even in the face of declining budgets.

For the richest nation on the planet, this is bizarre. The case for educational technology as an effective tool for learning is generally accepted. If we know these tools have value to learners, then we need to ask why we only reach 25% of our students at any given time. (Note: visits to actual classrooms confirm that many of them have little or no access to computers, so this graph only represents the national average.)

There have been research projects in many school systems devoted to one-to-one computing, and (depending on the design of the programs) some of these efforts have been more successful than others. These projects, though, are so small that their number barely affects the national average.

We talk about the educational computing "revolution" as if we have achieved something of great significance in the use of these tools to transform pedagogical practice. In fact, with a computer penetration this low, teachers can well be forgiven for continuing to teach in the same ways they did pre-computers. If only some have access at any given time, the chance to use technology to implement different models of classroom practice is blocked.

Suppose we grant the possibility that computers connected to the broadband Internet can be powerful tools in the hands of students (something I believe), then we should ask why we haven't made more progress, and, more to the point, why our progress has been brought to a standstill. In addressing this issue, I want to distinguish between universal access and one-to-one computing where each student has a laptop that affords true anywhere/anytime access to resources. I believe strongly that the one-to-one solution is the preferable approach, long term, but acknowledge that any solution that brings universal access is worth exploring. In other words, it is the educational value of access that is most important, not the specific form that access takes.

Why Did The Revolution Stall?

This question has several answers. The one that gets the most attention is cost. Technology proponents are often shut down by presumed budgetary considerations. "Yes, this is all a good idea, but we can't afford to spend \$400 per student (for example) for a dedicated computer." Statements like this are commonly made by the same people who gladly spend \$75 per textbook in each of several content areas per student – resulting in a per-student total that rivals that of a powerful computer with broadband access to the Internet.

One response to this criticism is to suggest that, if we didn't purchase traditional textbooks (or

Under \$300 for a new laptop plus a printer! Yes, this was a limited offer, probably selling at cost just to bring people to the store, but imagine what you'd pay if you ordered these by the thousands (or millions)! It is pretty clear that the barrier to universal access is not the hardware expense. Even if the hardware is upgraded every three years, given the trend in computer pricing, the replacement costs will be a fraction of what you'd pay today.

Does Hardware Even Matter Anymore?

On the surface this sounds like a silly question. Of course you need a computer if you are going to run software. This computer can be on the student's lap, or it can be a powerful system in the back office hosting applications being run by terminals in student hands. My personal bias is toward placing computers in the hands of students since most schools lack the communications infrastructure required to handle the volume of data required to serve the disparate needs of an entire school full of children all accessing information and software at the same time.

In fact, hardware does matter, and new developments in chip design are likely to pay handsome dividends in managing the computer power consumption and performance in schools.

While the trends of reduced cost and increased performance for processors will continue, this view, by itself, obscures other factors that deserve attention. The first of these is power consumption. Even in the absence of the "exploding battery" and "melting motherboard" problems revealed by several computer manufacturers in the past year, there is the critical issue of battery life for laptops, and school power needs for desktop computers. (Remember we are talking about computers for every child!) The assumption that increased power consumption comes with increased performance is understandable, yet flawed. There are a variety of ways to address power consumption in the design of the microprocessor ranging from circuit design to changes in the way chips themselves are fabricated.

AMD, for example, has incorporated power management into their microprocessor designs with an innovation called AMD PowerNow!™ technology (in the AMD Turion™ 64 mobile technology family of processors for laptops, for example (www.amd.com)). The technology automatically reduces power (and performance) when appropriate, and then kicks it up again when needed. This is a lot like the cruise control on most cars today that maintains a set speed, but automatically adjusts the gas sent to the engine based on road conditions. Gas is reduced when the car is on a level road, or going down hill, and increased when the car is climbing. In the computer world, entering data in a spreadsheet or word processor takes a lot less processing power than applying complex filters to a digital photograph with a graphics program. By reducing power when it can, and kicking it back up when needed, the AMD processor not only helps extend battery life, but allows for on-demand processing power when it is required.

In addition to changing processor speed on the fly, research is continuing on building microprocessors using a thin layer of silicon grown on top of an insulator. This work started in the 1970's, and is continuing to be advanced from the lab to the manufacturing floor. The reason this is interesting is because today's microprocessors are made from thin silicon films grown on top of special silicon wafers. These wafers have a thin barrier to keep current from draining from the grown film, but some current gets through anyway. More current leakage means more power consumption, resulting in many high performing computer chips needing special coolers just to keep them from overheating. By changing the substrate on which the microprocessor is built, power consumption is reduced, and battery life is increased even more.

To see how dramatic power reductions can be, note that the power consumption of a typical laptop computer is often well over 20 watts, constraining battery life to a few hours in most cases. This need not be the case, however, as evidenced by the use of the AMD Geode™ GX processor series, which can allow the total power consumption of a laptop to drop to less than two watts in some usage scenarios – less than one tenth the power consumption of a traditional laptop. This, no doubt, is the reason this processor was chosen for the One Laptop per Child (OLPC) project at MIT (wiki.laptop.org/wiki/Main_Page). The goal of this project is to design and build laptop computers specifically for children all over the world. These computers will cost about \$100 each, and include hand-powered generators for use in those areas where electricity is scarce or unreliable. The AMD Geode processor provides excellent performance at very low levels of power consumption making it an essential component of this computer. Far from trading cost for performance, the design of this processor (and other elements of the OLPC) shows that performance can be maintained even as cost and power consumption is reduced.

Another way to increase chip performance is to integrate more functions on the processor. Moore's Law states that the number of transistors on a chip doubles every 18 months. The question is not how to get more transistors on a chip, but what to use them for. If you've looked at a computer's motherboard, you see lots of chips present. The two big ones are the microprocessor and the graphics chip. But there are plenty of others – memory controllers, RAM, sound chips, modems, etc. Each of these support chips is mounted to the board and soldered in place. Since these chips are usually never changed over the life of the computer, there are tremendous efficiencies to be gained by putting as much functionality on the microprocessor as possible. (Many have speculated that this is one of the reasons AMD acquired ATI, the premier designer of graphics chips.) At the very least, memory controllers can be placed on the microprocessor which improves performance by not having to waste processor cycles moving commands back and forth to other chips. This is the approach AMD has taken with its Direct Connect Architecture and integrated memory controller – features of AMD64 technology which allow the processors, memory controller, and I/O to be directly connected to the CPU and to communicate at CPU speed. This level of integrated functionality is why the AMD Opteron™ processor was chosen for the Cray Red Storm supercomputer and it is being used in servers manufactured by HP, IBM, Sun, and other leaders in the market. While the AMD Opteron

processor (today) has two processor cores, it is only a matter of time before four or more cores are available, with the likely impact of increasing the overall performance tremendously over architectures that offload certain functions from the chip itself.

The time may come when an entire high-performance computer will consist of a single chip connected to a power supply, network cable, USB devices, display, and a hard drive. This lets the size and form factor of computers to be changed significantly from the boxes and clamshells we carry around with us today. Some of this exists today in the form of special function computers embedded in cars and other devices, but someday this approach will redefine the general-purpose computer as well.

Again, this approach is likely to save money and increase performance and reliability over the long term.

What About The Software?

But while it is true that hardware costs will continue to drop while the performance improves, the same cannot be said for software. While some providers of proprietary software bend over backwards to make their licenses affordable to schools, other vendors establish annual license fees that can make universal access virtually impossible to achieve.

Depending on the state, for example, Microsoft® charges an annual license for Windows® XP and Microsoft Office that can range to over \$100 per year, per computer. Even if the hardware was free, a state with a million students would find themselves strapped with an annual software license of \$100 Million, and this is just for the operating system and commodity software. Curriculum specific software or other tools geared specifically for education have to compete for software money being spent just to allow the computers to be legally turned on.

Clearly, it is the right of any vendor to set and charge any price they wish for their software. The marketplace will determine if the price-point is acceptable. If, however, you are interested in finding ways to afford universal student access to powerful computing, you need to think about alternatives that can save significant amounts of money.

Linux[®] On The Student Desktop

Fortunately, there is a solution to this challenge that can make software affordable even as the number of computers grows to fill in the remaining 75% gap in our schools: putting Linux on student desktops.

Linux is a UNIX-like operating system that runs on microprocessor-based personal computers. The normal operating systems on these machines today are Windows XP and the Mac OS[®] X. Like these, Linux allows computers to run software, access networks, communicate with other devices, and otherwise make it possible for your computer to do interesting things. While the original motivation for creating Linux was to bring a "serious" operating system to the personal computer, the Linux movement has grown to the point where it is fast becoming the dominant operating system for many applications.

Linus Torvalds, the originator of Linux, was a college student who first envisioned this as a project for his own use. He decided to make the source code of Linux available to the public at large, which had two effects. First, it built a community of Linux users capable of fixing errors in the operating system, and adding new features over time. Second, it insured that, as new computers and peripherals came on the market, Linux would be upgraded to run on just about any computer platform imaginable (including, for example, the iPod if you want).

Because it is based on UNIX, Linux is powerful, fast and reliable. Cray, for example, uses Linux as the operating system for the Red Storm supercomputer at Sandia Labs. Several U.S. telephone companies use Linux-based computers to manage telephone connections. Many of the commercial Internet service providers use Linux-based servers. Speed and reliability are critical in all these applications.

For years, Linux was largely relegated to the "back office," where highly technical folks accessed their Linux systems with text-based terminal programs reminiscent of the old days of Microsoft DOS. The casual user was generally too intimidated by the cryptic commands needed to install and launch software to take Linux seriously, so only a small number of dedicated users worked with this OS outside of specialized environments.

Recently, Linux distributions (distros) for ordinary computer users have been created that have a nice graphical user interface similar to those found on Windows XP or Mac OS X. When using one of these flavors of Linux, the learning curve for new users is amazingly short, as long as the user is already familiar with the Windows or Macintosh[®] operating system. The skills acquired in one system transfer to Linux just fine.

The ability to tailor Linux versions to specific hardware and to support a nice graphical user interface (such as Gnome or KDE) helps explain why MIT chose Linux as the operating system for the OLPC. Compactness and user-friendliness were both critical for this project.

The question is which version of Linux to use. A quick trip to DistroWatch (www.distrowatch.com) shows the download history for a hundred versions of Linux that are freely available.

The reason that Linux has been able to migrate and morph into different powerful versions is that the underlying programs are free to be examined, modified, and redistributed as the user wants. While there is a clearinghouse to insure that the Linux core remains stable, variations based on user interfaces, and accompanying software, proliferate.

This quality of Linux is important. Unlike Windows or the Mac OS which are "single source" closed operating systems, Linux users are free to choose the version that best meets their needs. This is important for school districts concerned about local control. There is no local control when using the Microsoft or Apple® operating system. The ability to choose from over a hundred Linux distributions brings true choice to educational computing for the first time. Of course this amount of choice can be intimidating.

Because of this, it is a good idea for a state or district to pre-screen various distros and prepare a list of recommended versions with enough background information to let schools make an informed choice. In this regard, the background of the provider is important. For example, you may want to have a real human being at the provider's company to help out. Maybe you need a specialized version pre-loaded with special applications for your school or district. And, just maybe, you don't want to take a chance with a vendor you never heard of before.

A Version Worth Exploring

All if these are splendid reasons to look into openSUSE™, and SLED (Suse Linux Enterprise Desktop), the Linux distros from Novell® (www.opensuse.org).

Novell has been in business since the 1970's and, among other things, acquired the license to UNIX from AT&T many years ago (since sold to another company). It would be fair to say that Novell has probably forgotten more about high-performance operating systems than most companies know. While Novell is probably best known for their back-office work, they are advancing their efforts in ways that impact student desktops, based on many years of deep experience in the K-12 education market. Furthermore, they are achieving this goal through the use of Linux.

With their extensive background in enterprise computing, massive networks, and other efforts, Novell's acquisition of the SUSE Linux project from Germany marked a change in direction toward open source development. The result, openSUSE 10, is a complete package that lets you install Linux and numerous core applications on your existing computers. If you don't need support (and are willing to download a DVD-worth of material from the Internet), you can get openSUSE for free (www.opensuse.org). More likely, you will want some support from Novell professionals, and for this the SLED offering is perfect. This version is available for a fee. Whatever you end up paying, it will be a tiny fraction of the money spent on your current software subscriptions.

A state with a million students could see their annual core software subscriptions fall from \$100 Million to near zero just by changing to this operating system.

While software cost is important, these savings would be meaningless if they involved a trade off in quality. In fact, there are several strong advantages in using Linux on student desktops (as well as on servers). First, Linux extends the life of hardware. Computers that run slowly under Windows XP become lightning-fast after moving to Linux. Cold boot times, for example, can drop by a factor of three by replacing a fresh copy of Windows XP with Novell's SLED. And the boot time for Linux does not creep up over time. One of my Linux laptops has been in active use for well over a year, during which time about a hundred programs have been installed and removed. The boot time on day one was 90 seconds, the same as it is today. (Note that for the Linux on that computer the time includes my manual log-in typing speed.)

Second, and amazingly important, I have yet to make a Linux system crash unless I did something amazingly ridiculous. My SLED laptop boots up every time, and it just keeps on working no matter how many applications or documents I have open at one time. Unless you log in to the root of the system and start messing around, you are free to do just about anything safe in the knowledge that the system will remain running. The importance of this factor cannot be overstated. The two most dangerous words in a technology-using educator's vocabulary are "watch this," spoken just before hitting a key on the keyboard. We joke that the reliability of a system is inversely proportional to the number of people in the room. I saw this in action a few months ago when I was on a panel with two other people. One was using a Mac, one was running Windows XP, and I was using Linux. We all had time pressures to get our computers swapped onto a projector before our respective presentations. The Mac user could not get the projector synchronized properly. The Windows user suffered a full system meltdown, and I just plugged my computer to the projector and saw the screen come to life immediately.

Teachers don't have time to mess with settings – to be useful in the classroom, computers need to work every time they are turned on. Period. The simple fact is that, even if it were more expensive than Windows, SLED would make sense based on reliability alone.

What About The Software?

As this is written there is one challenge to putting Linux on every student's desktop, educational software.

If you are new to Linux, you might well be asking, what about the software? Does Linux run Microsoft Office? Inspiration®? The iLife® suite? Does it run specialized educational software? What about Photoshop®?

The answer is that, as of this moment, most common educational titles are not shipping in versions that run under Linux. That said, there are two things you need to consider. First, in many cases, there are free open-source equivalents of commercial software, some of which are even more powerful than their proprietary counterparts. For example, OpenOffice (www.openoffice.org) performs the functions of Microsoft Office, even to the level of being able to import and export documents in the Microsoft format if you need to. Beyond providing this base functionality, OpenOffice also adds new features, including the ability to export a presentation as a Flash object for posting on the web, for example. And, OpenOffice is free. The GIMP performs many of the same functions as Adobe Photoshop, and is also free. The list goes on and on. The quest for powerful open source software was one of the reasons I wrote the book *When the Best is Free*, which describes several powerful programs that run well on Linux (and other operating systems, for that matter).

From the classroom perspective, your Linux provider can create special installer disks that pre-load your core open source applications along with the operating system so the computers are ready to use the minute they are turned on.

Because the number of free open-source titles for Linux is growing, the pressure is on for proprietary software vendors to create Linux versions so they can sell into this growing market before someone creates similar functionality and gives it away. Stagecast Creator (www.stagecast.com) and Visual Thesaurus (www.visualthesaurus.com) are just two examples of commercial software used in education that can run under Linux (as well as the Mac and Windows operating systems), and many more are on their way. The easiest way to get even more software converted is for schools and districts to contact the software publishers themselves and tell them that continued purchases will be based on their willingness to support Linux.

This is exactly what vendors have done in the past whenever Apple or Microsoft introduced a new operating system. Keeping up with operating systems is the price of admission for software vendors. That said, there are exceptions. Apple has little interest in porting their iLife suite to any other operating system since they use this software to help drive hardware sales. Other vendors only support one operating system because they lack the resources or interest to port their work to other systems. The reality in education, however, is that successful software

vendors need to have their software work on whatever computers schools have. The software experience needs to transcend hardware decisions whenever possible.

What future is there for commercial software when open-source software becomes more popular? The reality is that free software has always co-existed with commercial proprietary titles. People will gladly pay for quality and, if they don't have to pay for the operating system or commodity software, this leaves more money for specialized software. The open-source movement actually benefits several constituencies:

Hardware vendors can start to tap the 75% growth opportunity as scarce funds get redistributed, Subscription services (e.g., Safari, Nettekker) can reach more classrooms, Students gain the kind of technology access needed to develop and grow the skills needed to thrive in 21st century, Teachers get the technology support needed to teach in new ways.

The true beneficiaries are the creative fast-movers both inside and outside the classroom.

The Hidden Cost

So far we've explored the continued drop in the price of computer hardware, and showed how a powerful operating system and large libraries of commodity software can be obtained for free. There is one other cost factor that has not been mentioned – a cost that is inescapable. While it is true that a modern Linux implementation can be mastered by any current computer user in a short period of time, staff development is needed to help educators build the pedagogical bridge to learning in the environment of universal access. Unless many pre-service institutions change their practice, it is unlikely that they will be able to offer much help in this regard, which means that even the newest teachers will need assistance in learning how to help students achieve mastery with the aid of ubiquitous computing. Keep in mind that we all bring our personal experiences into the classroom, so if teachers lack the background in teaching with technology, teachers won't be able to transfer this enthusiasm to the students.

An Investment, Not A Cost...

While traditional face-to-face staff development will always have its place, much of it can be supplemented with on-line activities where teachers and educational leaders get to acquire new skills using the same tools their students have. This investment needs to be thought of in a broad context. If each teacher impacts the lives of thirty to 150 students per year, and teaches for two to three decades, the impact of effective, ongoing, and well-supported staff development gets multiplied by the benefit it brings to 600 to thousands of students over the career of each teacher.

As I've said for many years, in the absence of powerful transformative staff development, the dominant impact of technology in the classroom is likely to be an increase in the electric bill.

Yes, the promised computer revolution in education is missing. We have only achieved 25% of the goal; but the remaining 75% is achievable if we are willing to take advantage of the heroic efforts of those creative people and organizations who have dedicated themselves to making computing accessible to anyone, anytime. To make this happen, alliances are essential. Everyone related to education needs to be committed to the vision and to making it happen. We need to support student learning by any means possible as part of a global community.

All of us, pulling together, can help every student in the country be all she or he can be. Let's unleash this power, and let's unleash it now.

About The Author

Dr. David Thornburg is founder and Director of Global Operations for the Thornburg Center, an international consulting firm specializing in K-12 education and the role technology can play in support of student learning. Prior to founding the Thornburg Center in 1980, Dr. Thornburg was a Principal Scientist at the Xerox Palo Alto Research Center where, among other things, he invented user interface technologies for personal computers still in use today.

*Dr. Thornburg is an active proponent of Linux and open-source software in general. His book on open-source software for all computer platforms, *When the Best is Free*, has received accolades. For more information on this book and other topics relating to this briefing, visit www.tcpdpodcast.org. He was chosen to take part in the high-level K-12 Open Technology Summit in Raleigh, NC.*

David gives presentations on topics related to educational technology throughout the world, and is often on the programs of major educational technology conferences in the US. His pedagogical interests lie in the constructivist domain of inquiry-driven project-based learning, and he believes that technology needs to be applied in ways that honor the integrity and capacity of every child as a lifelong learner.

David's workshops and other staff development activities can be secured through him directly. He loves hearing from his readers. E-mail him at dthornburg@aol.com