Improving Education – Linux and Open Source

An Alliance Technologies White Paper by Josh More

Introduction

Linux is an alternative operating system, comparable to Windows, that is designed to run as stably and efficiently as possible. It is one of many *open source* applications – freely modifiable projects designed as a community effort to benefit the world as a whole. While originally an experimental project, today the Linux operating system runs super computers, Internet and business servers, workstations, desktops, laptops, and even cell phones. It is being used, in conjunction with other open source software, to help people run more securely, efficiently, and flexibly then ever before. The union of Linux and open source software is used by businesses, governments, militaries, and schools all over the world. This paper discusses how the two can improve flexibility in education and reduce costs. Particularly:

- Students can improve their understanding of technology, not isolated applications.
- Schools may choose systems matching their philosophies, preventing vendor lock-in.
- Students can do more meaningful work, allowing them to move beyond rote exercises.
- Reduced costs have allowed schools numerous opportunities:
  - to budget for teachers and supplies instead of software licenses.
  - to get new use from older hardware.
  - to allow more systems to be supported by the same staff.
  - to provide experimental systems for projects that otherwise would be too expensive.
- The security in Linux makes it far more resistant to viruses and worms than Windows.
- Similar systems could be used at home, providing a consistent learning environment.

Historical Overview

Over the last decade, the presence of computers in American schools has expanded dramatically. On one hand, this shows that schools are recognizing the importance of teaching students about technology. However, it is disheartening that many of these technology programs focus more on training children how to use specific applications rather than promoting understanding and working with various technologies. In other words, computers should be used to teach analytical skills, the application of logic, and critical thinking.

The power of computers is in the ability to adapt them to the needs of the user. When the user is forced to adapt to the machine, they tend to distrust and dislike the technology rather than appreciate the flexibility it can bring to various tasks. Societally, this has resulted large numbers of people self-identifying as "computer illiterate". The prevalence of this term in common parlance promotes the acceptability of illiteracy in our society.

Moreover, this trend is increasing. Due to the recent increase of security problems and license costs, the *standard image*, a set of applications beyond which a computer may not vary, has been growing steadily more repressive. A standard image is increasingly limiting the number of applications installed, often to a single application per specific category. While this level of restriction makes sense from a business and management perspective, its presence in schools results in a limited and frustrating learning environment. Additionally, when one compares the applications in a standard image from ten years ago (Lotus 1-2-3, DBase, Gopher, Word Star, DOS . . .) to those that graduating students use today, it becomes quite apparent that the application-training approach is unwise, and that any useful conceptual learning occurs incidentally.

On the positive side, a social trend has been emerging to combat these restrictions. Linux and open source are returning freedom to schools everywhere. Combining the efficiency that computers bring to common tasks, the flexibility that they offer to creativity, and the global user communities, this system allows students to be taught the concepts behind the applications rather than just being trained to push a button.

Philosophies

While the conclusions herein should be valid for most educational systems, this paper focuses on the Montessori method of education. Philosophically, the Montessori approach treats children as individuals who should be allowed to discover new understandings at their own pace, while being guided by an adult. This is done by engaging all of the child's senses in the learning process and treating the children with trust and respect so that learning may happen naturally. It is an approach to education based on preparing the environment and guiding the child rather than forcing the child to conform to arbitrary standards.

The Montessori approach of a structured learning environment suggests that great care should be taken before introducing something new to the classroom. Given the complexities of computers and the restrictiveness of *standard images*, it is unsurprising that acceptance of the technology has been slow. However, the fact remains that some children have access to such technology at home, and are expected to use it in high school and college, so some formal education would be desirable. In effect, schools today find themselves in a similar position to where schools were when the Montessori method was initially being refined.

Ideally, one would have a customized system for each Montessori age cluster with specific tools installed and configured such that children may use them to learn in the most appropriate ways. Additionally, the tools could be chosen such that they could be used through hands-on manipulation, thereby reducing the abstraction of computer use as much as possible without diluting the power that such abstraction may bring.

With a Linux/Open Source system, a school can not only help the students to learn, but give the students the opportunity to teach and to do meaningful work. Many open source projects exist in the world that accept contributions, be they enhancements to code, documentation, or clip art. The support and development communities can provide children with the ability to benefit others through their daily work – at whatever level the child prefers. As these projects exist to benefit everyone, rather than to bring profit to a specific company, supporting these projects can be viewed as a more ethical use of public funds. More locally in scope, older children can help adjust the system images used by the younger children, thereby streamlining the management of the systems.

Practicalities

While philosophies are important, from a practical standpoint, computing deployments generally come down to three points: cost, maintainability, and flexibility.

Looking first at the issue of cost, there are many that say that Linux is free. This is partly true. While there are truly no license fees involved, there are often costs for hardware and software maintenance. Luckily, many schools can benefit from many programs to obtain free computers. Also, as Linux runs surprisingly well on older hardware, schools can often reuse existing machines. Additionally, the sponsorship of a company can drastically reduce the costs of maintenance, as it can sometimes be worth the expense to a company to provide IT to a small school in return for the press it brings in as well as the opportunity to test out newer technology in a live environment.

Some larger schools have chosen to take on the burden of managing IT, rather than seeking out a corporate partner. Even in these cases, they have found substantially reduced costs: Riverside High School (California) saved $40,000 per year and The Multnomah Education Service District (Oregon) saved $200,000 per year by deploying Linux internally. The project was so successful, spokespersons for both organizations have taken to saying: *"Don't buy software, buy a teacher."*

Next, consider the maintainability of Linux and open source systems. People are often surprised to find that they are actually easier to run and maintain than Windows equivalents. The initial transition is also easier that many expect. The stronger security architecture of Linux makes it far more resistant to viruses and worms than Windows, and the ability to update most of the system without reboots means that security patches can be deployed more quickly without as much concern for system stability. There also exist several active communities to help schools work with Linux, which can dramatically aid in helping schools administer their systems. Moreover, in partnership with a company, the responsibility of maintenance can be completely taken away from educators, allowing them to focus on teaching and improving their professional skills. Better still, given the nature of Linux, should a partnership fall through, taking back responsibility for administration tasks or finding another partner is not difficult to do.

A final point in Linux's favor is its flexibility. A well-chosen Linux system includes a wide variety of software that is appropriate for children. Many educational games, even in the Linux world, are structured from a reward/punishment perspective. The advantage of Linux here is that such systems can often be modified to remove any features that may be incompatible with the school's teaching principles. Similarly, word processors and other productivity tools aimed at adults often have more complexity than children need. These systems can be modified to reduce complexity according to the target age of the child.

Also, you have the power to choose from many different applications. An exact list of applications that best fit a classroom would be highly dependent on the teacher, but the list that follows are of some of the more popular choices:

Dr. Geo allows students to learn geometry through interactive manipulations of shapes and relationships. It can also be scripted to explore more advanced topics in mathematics and physics.

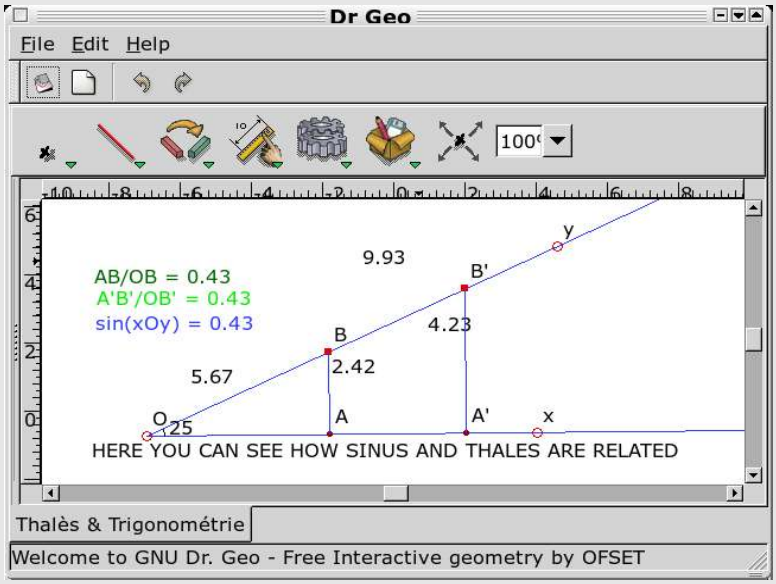

Illustration 1 - Dr Geo. (http://www.ofset.org/articles/15)

KDE-Edu includes many applications:

- **Logo** – a programming language aimed at helping children learn programming and shapes through drawing techniques.

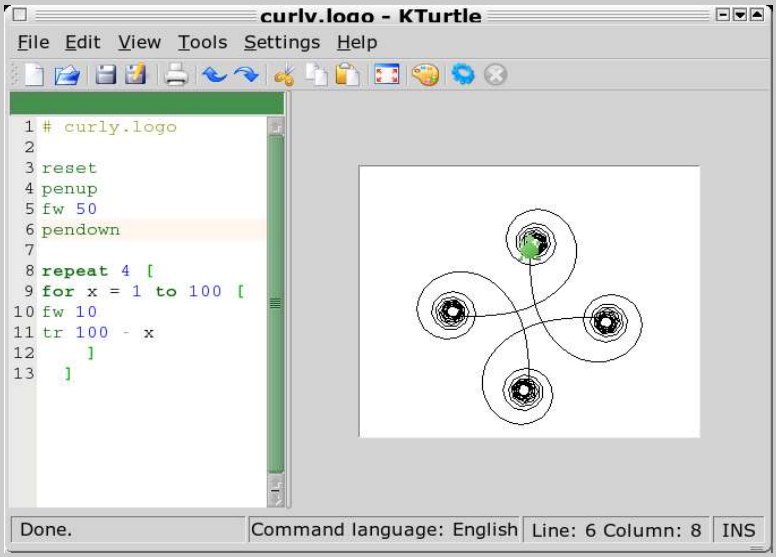

Illustration 2 - KTurtle (http://edu.kde.org/kturtle/)

- **Planetarium** – an application that simulates the night sky at different times and places. It allows the viewer to be located anywhere on the Earth. Labels and objects for stars, planets, and deep space objects can be conditionally displayed. It can also be extended with various star catalogs.

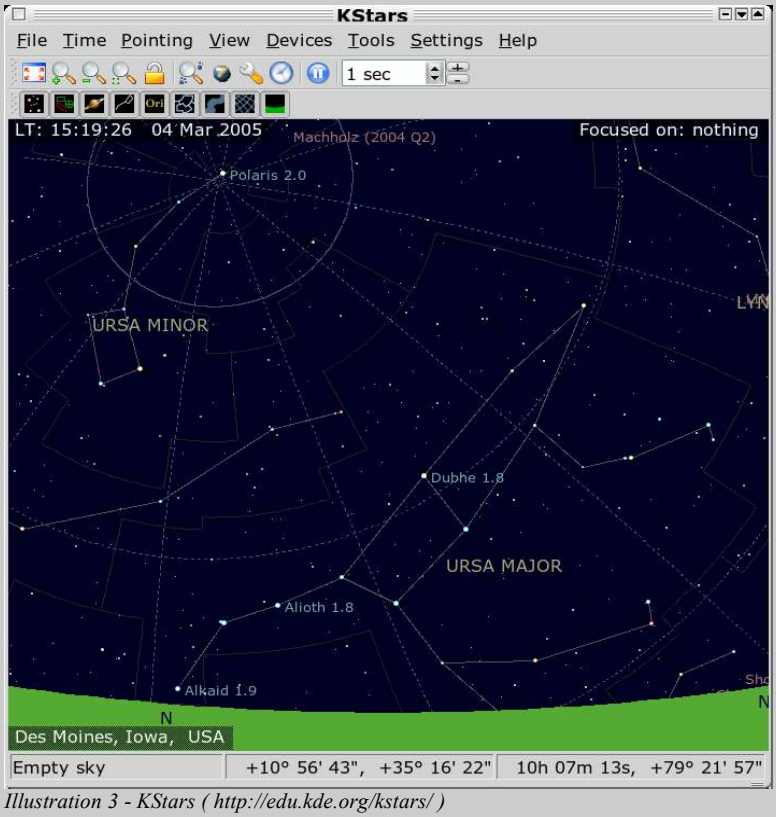

Illustration 3 - KStars (http://edu.kde.org/kstars/)

- **Periodic table** – an application that allows children to manipulate the periodic table and explore properties of different elements. It can display the relative states of matter depending on varying temperature, provide scatter plots of statistical data, display the elements by properties and historical views. It also provides specific, detailed, information regarding every element.

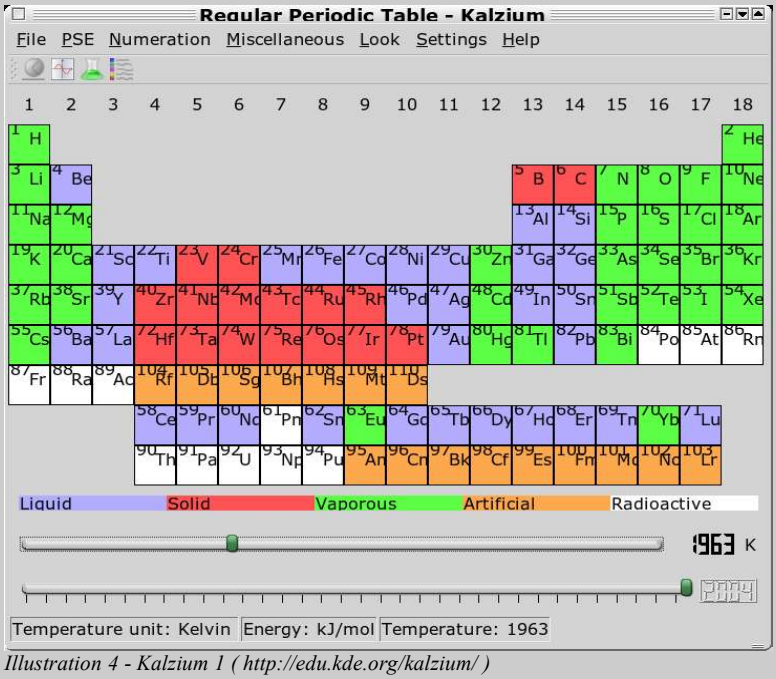

Illustration 4 - Kalzium 1 (http://edu.kde.org/kalzium/)

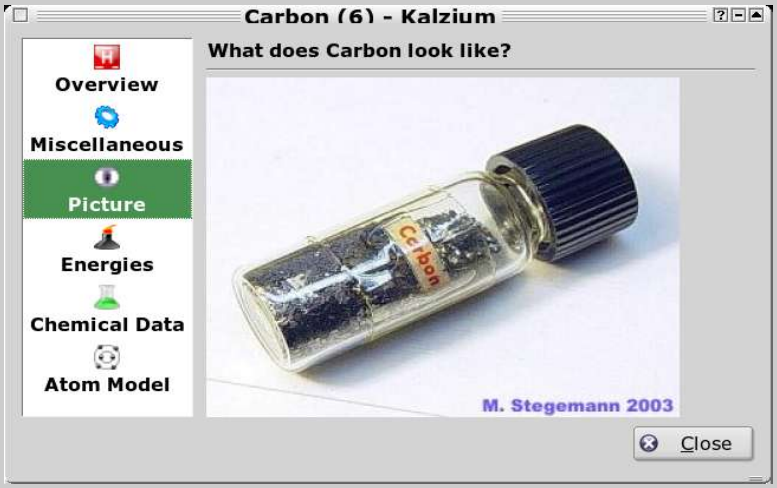

Illustration 5 - Kalzium 2 (http://edu.kde.org/kalzium/)

- **Flash cards** – an application that allows children to create and drill with flashcards for different topics. It includes the ability to stress the topics that the child does not know as well, to increase the absorption rate. It also provides various self-quiz styles so the child may choose the method that best fits his or her learning style.

Illustration 6 - KWordQuiz 1 (http://edu.kde.org/kwordquiz/)

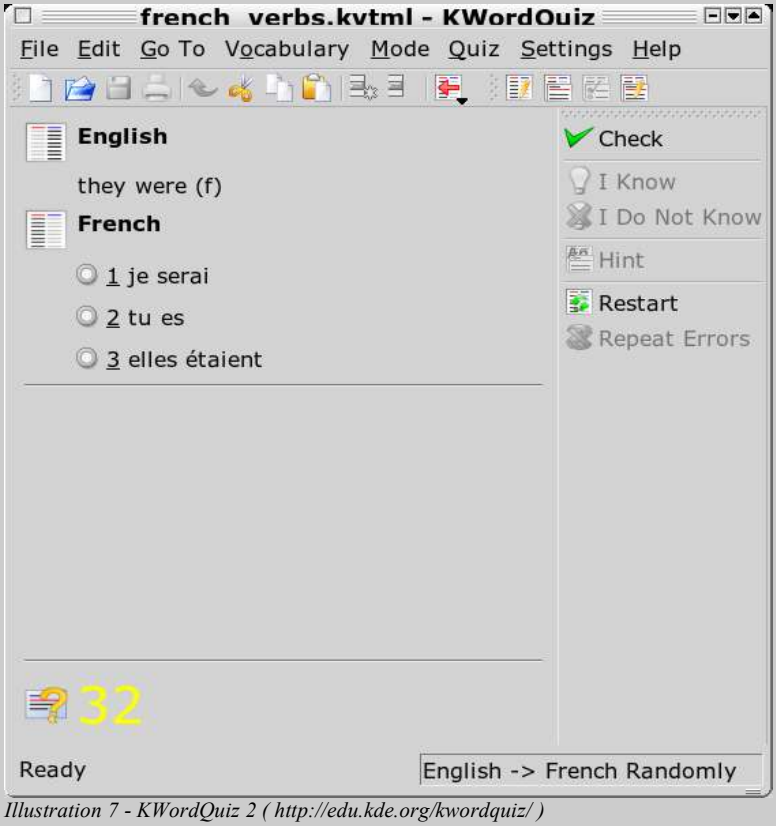

Illustration 7 - KWordQuiz 2 (http://edu.kde.org/kwordquiz/)

- **Language tools** – various tools to help learn Spanish, Latin, French, German and Japanese, with more languages coming soon. Methods range from simple vocabulary drills to more complex grammar guides. Additionally, the entire system can run in different languages, allowing students to learn through immersion.

Illustration 8 - KLatin (http://edu.kde.org/klatin/)

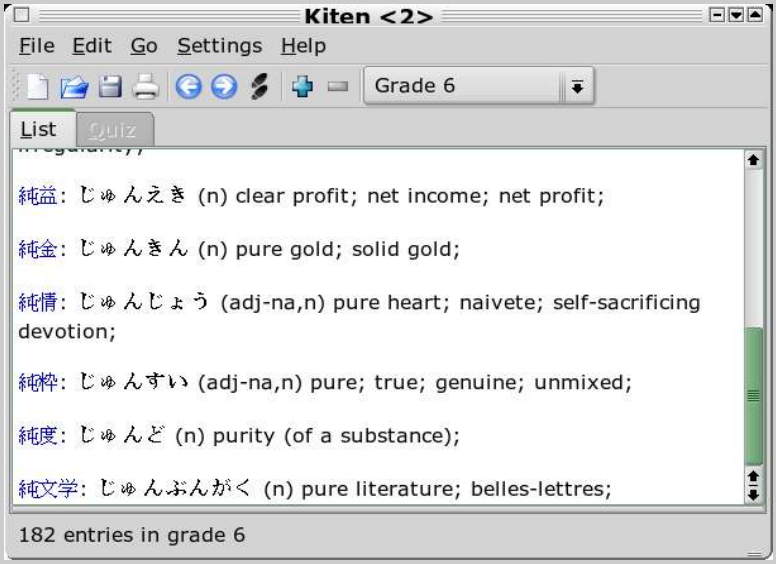

Illustration 9 - Kiten (http://edu.kde.org/kiten/)

Gcompris is a French-originated system for young children that includes counting, telling time, learning the water cycle, letter recognition, reading, drawing, and animation tools, among others. It is also intended to be easily extended. Older students could make learning tools for younger students, in their school as well as around the world.

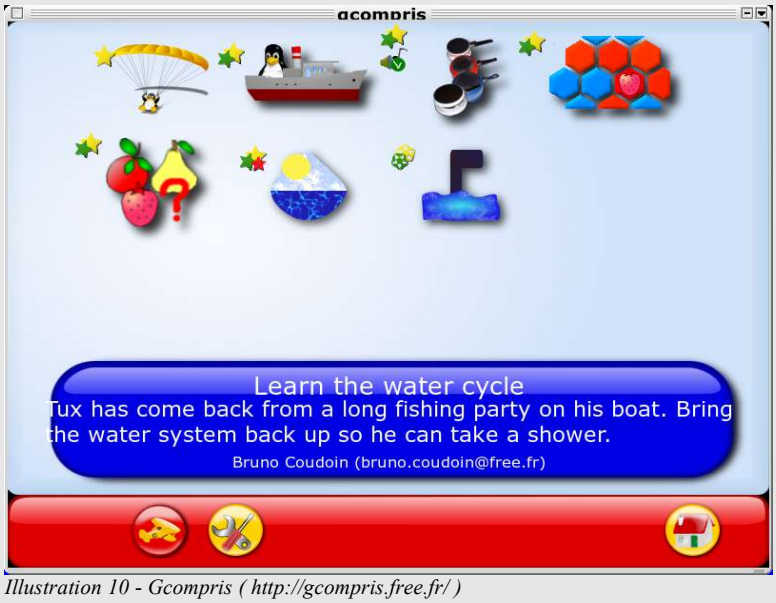

Illustration 10 - Gcompris (http://gcompris.free.fr/)

Tuxpaint is a graphics application that is aimed at allowing kids to be creative with images. It provides a simple interface but still teaches children about graphic manipulations including raster and vector images, text options, image "stamps", and transformation filters.

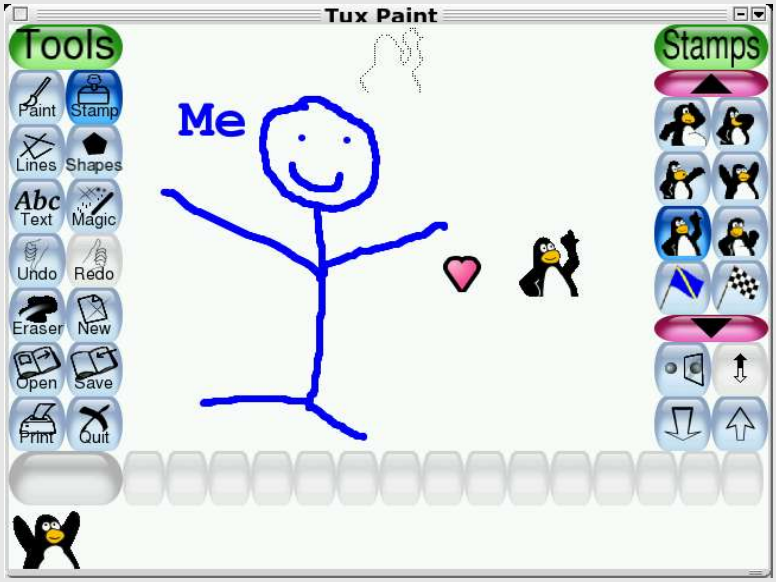

Illustration 11 - Tux Paint (http://www.newbreedsoftware.com/tuxpaint/)

Povray, **blender**, and **tulip** are just some of the 3D applications available, and allow kids to explore with mathematically generated shapes, both creatively and for scientific reporting. In addition to teaching measurement, three dimensional geometry, and optics, these systems can also be used to create photo-quality artistic images and movies.

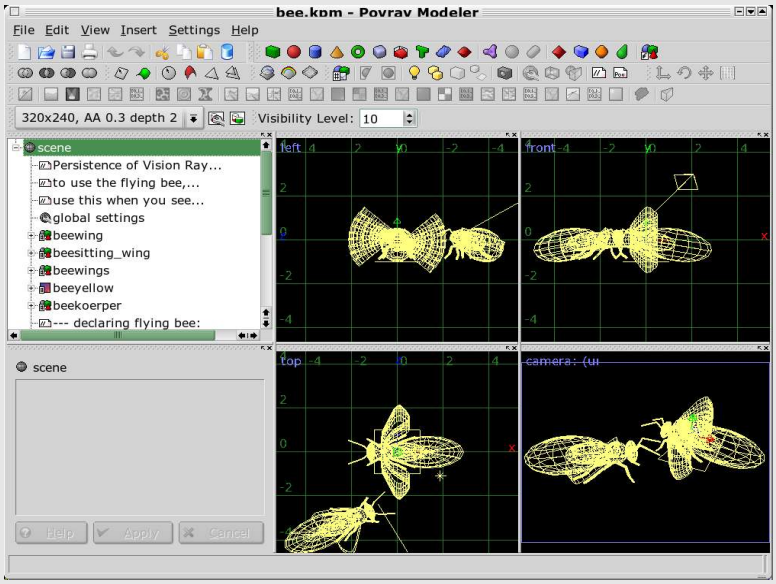

Illustration 12 - Povray 1 (http://www.povray.org/)

Illustration 13 - Povray 2 (http://www.povray.org/)

Music composition, ear training, and performance tools such as **Rosegarden**, **Solfege**, **Noteedit**, and **Montessori Bells** are also included. These can provide tools for students to do true music composition without necessitating the existence of an orchestra or band. It can also provide an environment for a child to explore sound without the concern of embarrassment if something does not sound as they expected it to.

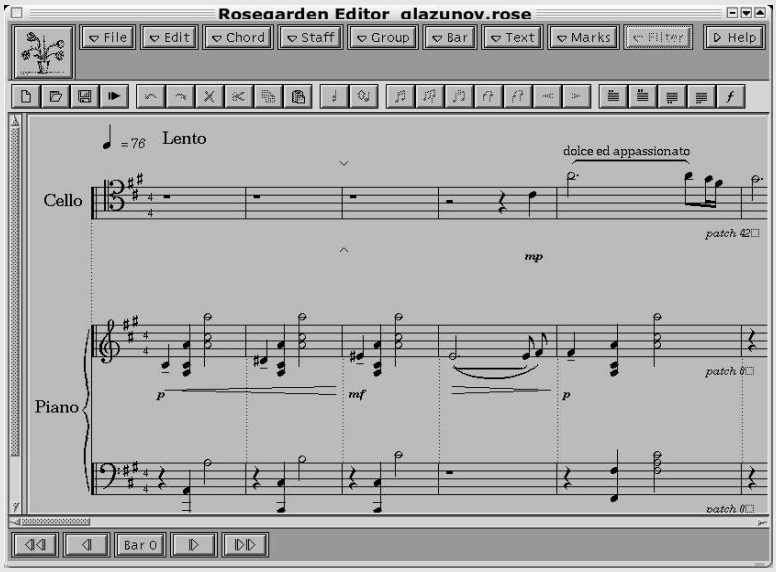

Illustration 14 - Rosegarden (http://www.rosegardenmusic.com/)

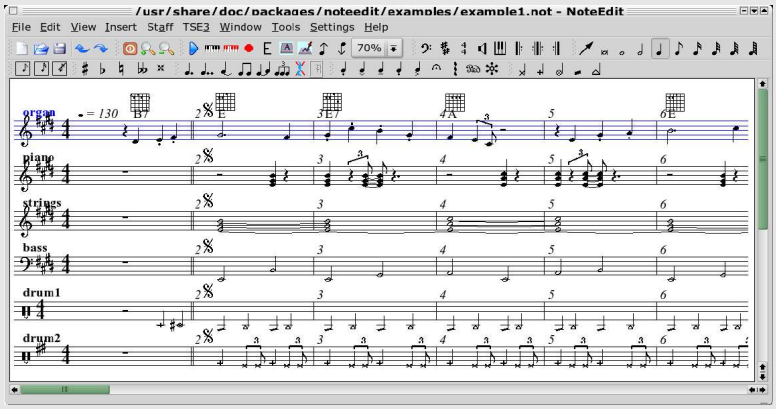

Illustration 15 - NoteEdit (http://noteedit.berlios.de/)

For children learning research papers or other projects requiring self control and time management, the project management tool **Planner**, the brainstorming tool **View Your Mind**, and the flowcharting tool **Kivio** can be useful.

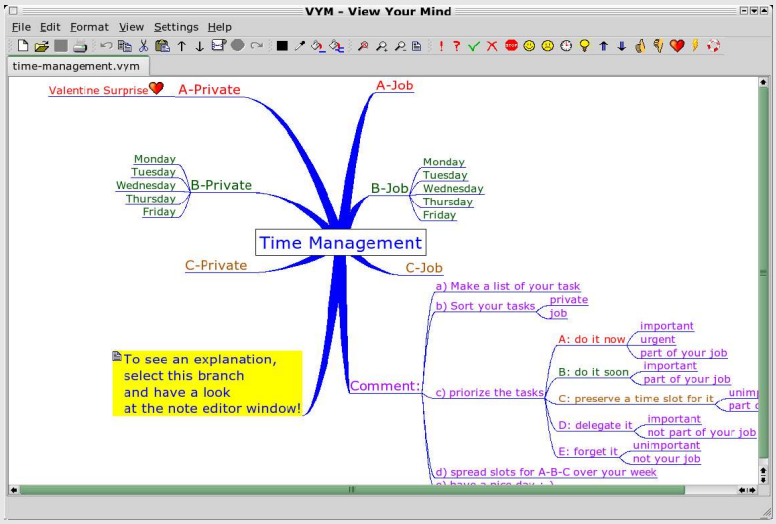

Illustration 16 - View Your Mind (http://www.insilmaril.de/vym/)

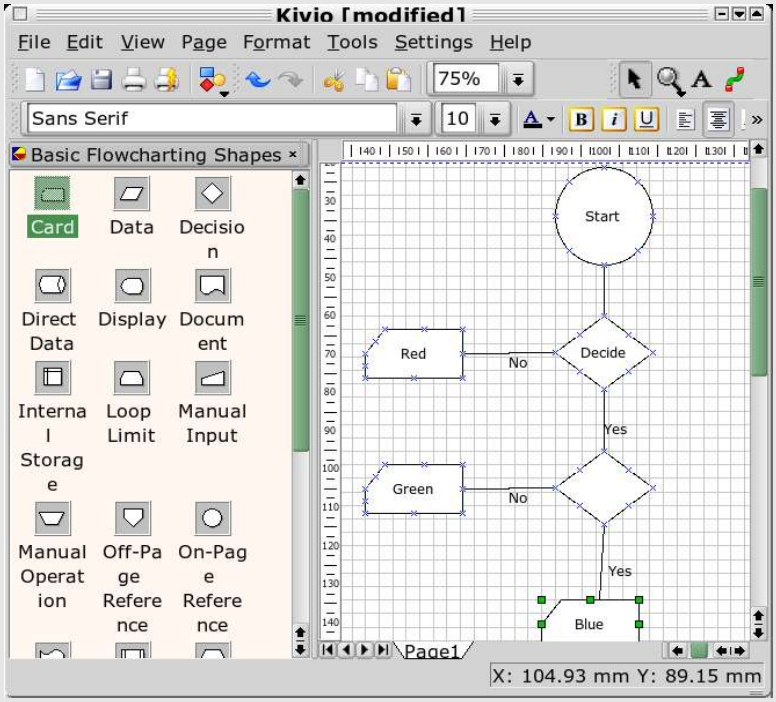

Illustration 17 - Kivio (http://www.thekompany.com/projects/kivio/)

The word processor **OpenOffice.org** can do anything required for papers and presentations. Additionally, it is capable of offering a simplified interface so as to make it less overwhelming, thereby allowing children to focus on work instead of playing with the application.

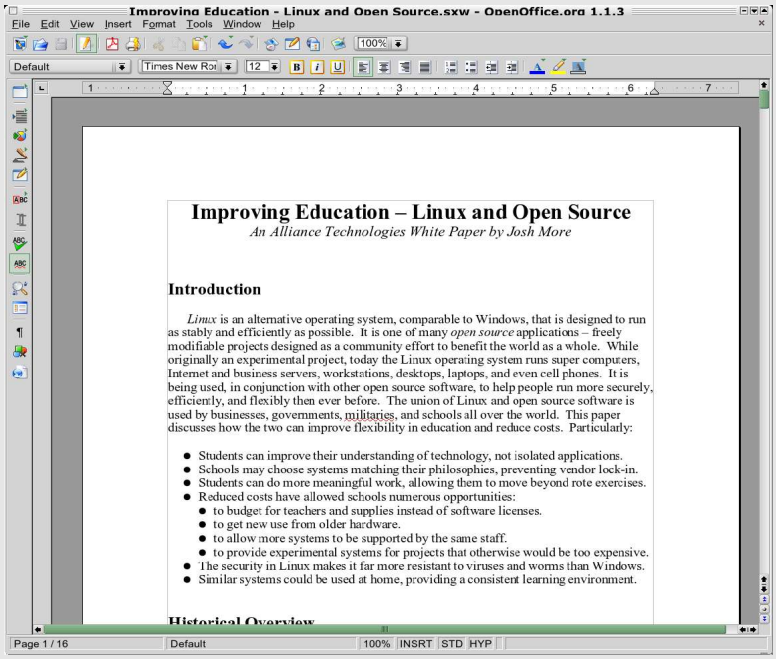

Illustration 18 - OpenOffice.org (http://www.openoffice.org/)

The web browser **Firefox** natively offers tabbed browsing and pop-up blocking, but can also be extended to block advertisements and offer enhanced research functionality.

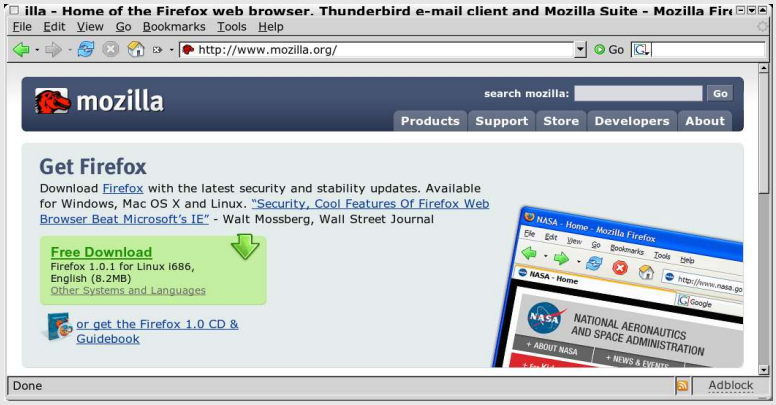

Illustration 19 - Firefox (http://www.mozilla.org/)

Professional programming languages such as **C**, **C++**, **Java**, and **Perl** are native, though the also included child-friendly languages such as **Logo** and **Squeak** are recommended for the younger computer-inclined students.

As all of these applications are free, a child could also use them at home without licensing costs, though to do so may require some parental expertise. In some cases, a partnering company could arrange to support similar images for use outside the school, so that a child could experience an identical computing environment wherever they may be. While this shifts the child's exposure to other technologies to the parents, it also provides the children with an opportunity to show and teach adults what they are using in school, as well as engage in more authentic tasks. This multi-generational learning and teaching experience may server to reinforce any concepts being taught as well as improving the child's self esteem.

Conclusion

A computing deployment based on Linux and open source could provide benefits that Windows cannot match.

- In many cases, Linux is less expensive to both purchase and maintain.
- There may be opportunities for a school to partner with a corporate sponsor which could result in even greater cost savings than handling maintenance internally.
- Outsourcing the maintenance of computing systems also provides a more concrete focus on the teacher's role of educating, thereby helping to professionalize the industry.
- Linux has a wider variety of configurable applications in many different categories.
- There is no vendor lock-in. If a vendor or application is not operating correctly, the open source nature of the solution allows it to be easily modified or replaced.
- Linux and open source solution are capable of providing better learning experiences by allowing the children to do work more effectively while simultaneously bypassing the negatives of training in applications.

For these reasons, it is in the best interest of schools to investigate Linux for their computing needs.

Resources (links accurate as of March 4th, 2005)

Gebhardt-Steele, Peter. "The Computer in the Montessori Class" *Montessori Congress 97.*
< http://www.ilu.uu.se/ILU/MONTESSORI/MThecomp.htm >

Valentin, Eva. "Work with Computers for 4-9 Year Old Children" *Montessori Congress 97.*
< http://www.ilu.uu.se/ILU/MONTESSORI/MWork.htm >

Case Studies Database. Ed. info@schoolforge.net
< http://casestudy.seul.org/cgi-bin/caseview0.pl >

K-12 Linux in Schools – K12LTSP – Case Studies. Ed. Lincoln D. Stein.
< http://www.k12ltsp.org/casestudy.html >

Computers at Jarrow. Ed. Jarrow Montessori School.
< http://www.jarrow.org/programs_computersat.html >

Computer Policy. Ed. Roots and Wings - Montessori Primary/Elementary School.
< http://www.rootsandwingsbc.com/pages/philosophy/computer.html >

Technology at Ruffing. Ed. Ruffing Montessori School.
< http://www.ruffingeast.org/webpages/technology.htm >

Overview of Technology and Education Reform. Ed. US Department of Education.
< http://www.ed.gov/pubs/EdReformStudies/EdTech/overview.html >

Statement of Basic Principles and Suggested Actions. Ed. Society for Information Technology and Teacher Education, 1998. < http://www.aace.org/site/SITEstatement.htm >

Critical Issue: Promoting Technology Use in Schools. Ed. Jan Gahala, M.A.
< http://www.ncrel.org/sdrs/areas/issues/methods/technlgy/te200.htm >

Critical Issue: Developing a School or District Technology Plan. Ed. Dr. Larry S. Anderson.
< http://www.ncrel.org/sdrs/areas/issues/methods/technlgy/te300.htm >

Open Source Software in Schools - A New Paradigm. Ed. Paul Nelson.
< http://k12os.org/k12linux_casestudy.doc >

Montessori Education: Philosophy, Characteristics, and Frequently Asked Questions. Ed. Lake Forest Park Montessori. < http://www.lfpm.com/montess.html >

Montessori Overview. Ed. Montessori Country School.
< http://www.montessoricountry.com/montoverview.htm >

Montessori Training. Ed. North American Montessori Center.
< http://www.montessoritraining.net/what_is_montessori/overview.htm >

The Montessori Method for 3-12+. Ed. Michael Olaf.
< http://www.michaelolaf.net/1CWhome.html >

creative commons

COMMONS DEED

Attribution-NoDerivs 2.0

You are free:

- to copy, distribute, display, and perform the work
- to make commercial use of the work

Under the following conditions:

Attribution. You must give the original author credit.

No Derivative Works. You may not alter, transform, or build upon this work.

- For any reuse or distribution, you must make clear to others the license terms of this work.
- Any of these conditions can be waived if you get permission from the copyright holder.

Your fair use and other rights are in no way affected by the above.

This is a human-readable summary of the Legal Code

(http://creativecommons.org/licenses/by-nd/2.0/legalcode).

(Creative Commons Disclaimer: http://creativecommons.org/licenses/disclaimer-popup?lang=en)