

GOVERNMENT SPONSORED OPEN SOURCE SOFTWARE FOR SCHOOL EDUCATION

Andrew E. Fluck

University of Tasmania, Australia

E-mail: Andrew.Fluck@utas.edu.au

Abstract: The history of computing can be seen in terms of cultural change engendered by philosophical opposition. In education some key philosophical dichotomies are those of commercial-copyright versus free open source software, and office-centric pedagogies versus rich software libraries to support the whole school curriculum. The 'Blue File' software library was generated by four centres supporting students with special learning needs in Britain in the early 1980s, and embodied important features relevant to these tensions. More recently a longitudinal study in Tasmania has provided data to take the story further, highlighting the dangers of an office-centric approach to ICT in school education. The paper concludes with an argument for the exploration of online learning objects as a possible teacher-friendly solution to providing teachers with a rich repertoire of curriculum-focused learning activities.

Key words: Free source software; Open source software; School education; On-line learning objects; Office productivity software.

1. INTRODUCTION

The history of computing is rather like that of nations: a continuous series of battles arising from differences of belief. This paper concentrates on two entwined differences of philosophy. The first is the battle between proprietary and open standards. The second is the battle between office-centric conceptualisations of ICT in school education and the power of rich libraries of curriculum-focused software. The outcome of these battles has a great deal to tell us about management of a change process that requires and will enact fundamental cultural shifts.

2. GOVERNMENT-SPONSORED SOFTWARE

In the early 1980s the British government supported the introduction of ICT into schools by subsidising their purchase of the endemic BBC microcomputer. This model achieved export status and entered a second generation with the Archimedes range before obsolescence beneath the heels of the PC. Government support for educational software and teacher professional development was provided through the Micro-Electronics Programme that set up 14 regional centres for general education and four others to support children with special educational needs. These four SEMERCs (Special Education Micro-Electronics Resource Centres) were managed by Bob Dyke/Martin Littler (Manchester); Jean Tait/Andrew Fluck (Redbridge, London); Jean Johnston (Bristol) and Colin Richards/Derek Cooper (Newcastle). They were established in 1984 under the auspices of the Council for Educational Technology with Mary Hope as national co-ordinator (Jean Johnston in Vincent, 1989, p.276 & Saunders, 1984, p.95). Most were closed in 1989.

These four centres were responsible for the development of a range of software that was freely copyable, called 'The Blue File'. The writing of this software was promoted by encouraging and sharing ideas between teachers, commissioning programmers to implement software suggestions, providing release time for teachers to write programs themselves and by centre staff taking on all these roles. This range of software was primarily intended to be used by children with special educational needs. The programs therefore addressed basic skills such as matching, selecting, simple numeracy, switch-operated communication etc. Making the programs simple to operate for these students yet catering for a range of input methods to enable those with physical disabilities to use the software, represented a real challenge. Some of the most advanced technologies of the time were utilised to address these difficulties. Input devices such as the expanded keyboard, the photonic wand (a light-pen equipped with a small telescope was mounted on one arm of some eye-glasses), touch screens, all kinds of switches, and in particular the Concept Keyboard, were used. This latter was a touch sensitive tablet upon which a sheet of paper could be placed carrying pictures or icons representing elements of a learning situation. When connected to a simple word processor, students could write sentences by merely touching pictures or phrases on the paper as it rested upon the concept keyboard. Output devices such as robotic devices, turtles, musical instruments and voice synthesizers were also employed.

Some of the key aspects of this software are recognised as important today, such as their framework or content-free nature, the customisability of tools and toolbars, and the capacity to adapt to user-defined input devices.

This was recognised as vital to accommodate the range of intellectual and physical abilities of the children with special needs, and this kind of accessibility is now built into operating systems. Another important aspect of this software library was its breadth of curriculum coverage, offering simulated canal locks, activities to teach life skills and tools to promote literacy through interactive scene construction.

The 'Blue File' programs were freely copyable for educational use in the United Kingdom and were distributed around the world, even in Western Australia (Morgan, 1990 ,p.14). Most were for the Acorn BBC computer, but many were later transferred to the PC platform. Over thirty titles became available, as described in Table 1.

Table 1. Blue File software descriptions

<i>Title</i>	<i>Description</i>
Acorn Speech programs	Six simple programs incorporating spoken instructions from the voice output hardware.
Annual Review System	Database for tracking individual students through personal curriculum plans.
Beans on Toast	A sequencing simulation for cooking a meal using a gas cooker with rich graphics and animations.
Belmont Programs	Nine programs covering colouring, matching, money and sequencing for children with severe learning difficulties.
Berkshire Catalogue	Software database which indexes programs according to their curriculum area and age level.
Brian Apter programs	Visual stimulus programs for young switch users.
Compact 1+2+3	These programs aim to stimulate language by giving a wide range of simplified scenes controlled by switches, the keyboard or the concept keyboard.
Driver	Simulation of driving a car, using the concept keyboard as an input device for the steering wheel, gear stick and foot control pedals.
Early Fingers	Targeting etc using a multiple-point infra red scanning touch screen from the Spastic Society. The programs feature animated cats and mice and cover sequencing, targeting, tracking, early number concepts and painting.
ECL	A collection of software for the Tandy book and concept keyboard on counting, measuring and matching.
Eresby software	Multiple input device programs (QWERTY or concept touch sensitive keyboards, or sound sensitive micromike) on dressing, counting, memory, and drawing.

<i>Title</i>	<i>Description</i>
Firs	Early learning material covering counting, subtraction, coins, sorting words and for preparing arithmetic worksheets.
Floor Robots	Pre-LOGO activities controlling a turtle using joysticks, microphones and other devices.
Inner City	For reinforcing basic skills in primary and special schools, with several simple adventure games.
Lists	Jargon-free database for very young children to explore categories of self-concept.
Look & Think	For visually handicapped youngsters.
Magic Seeker	This contains a set of data files and overlays to go with a range of content-free software such as Prompt, Touch Explorer and Lists, concentrating on the concept of a magical journey of discovery.
MAISE programs	Eight programs covering counting, words and shape-matching.
Picture	An early pre-number game for young children with severe learning difficulties, linking the screen display with pictures on the concept keyboard.
Moving In	A simulation of unloading a removal van full of furniture into a house. Control is through creating grammatically correct sentences using QWERTY or concept keyboard.
Picture Play	Picture-making activities using a range of input devices.
Prepositions	Practices the use of prepositions (in, by, beside, underneath, behind etc) with simple animation effects.
Programmer's Toolkits 1 & 2	Early utilities for freezing processor activity to facilitate screenshot capture and printout. Also includes routines for image compression.
Prompt 2 & 3	Flexible and popular assistive word-processor which allows students to use the QWERTY or concept keyboard to input text.
Redfile 1 & 2	Weaving simulation, building sentences by numbers, visual stimulus, early number and attention span software using single switch or keyboard input.
Simple use of language	The antics of Fred the fly: this figure is animated to practice use of nouns, adjectives and prepositions.
Single Switch Trainer & Wiper	These two programs enable a child using a single switch to learn how to gain control with it, building up a scene.
Sums	Drill and practice software for exercise on the four rules of number. Problems are generated automatically at set levels of ability, and help/explanation is available at every stage.
Touch Explorer	Classic content free software for concept keyboard touch-sensitive pad. Teachers can link pictures or real objects to text and voices.

<i>Title</i>	<i>Description</i>
West Midlands	Early learning programs.
Wordsearcher	Enables children to build their own simple word-puzzles and have the printouts in double-height characters.

Learning to Cope, 1986/7

3. OPEN VERSUS PROPRIETARY STANDARDS

In retrospective review this range of government-sponsored software was critical to the adoption of information and communication technology (ICT) in not just special education but the broader school education community. One example can be drawn from the first version of the National Curriculum for England which suggested students should be able to “construct a simple story as a sequence of words pictures or sounds using an overlay keyboard or mouse” (HMSO, 1990, p.45). Eight years later the training standards for teachers required them to know the characteristics of talking word processors (Department for Education and Employment, 1998) an early example of which had been pioneered in the ‘Blue File’ software. These examples of non-traditional input and output systems were only one aspect which continued in later educational practice. Another was the way in which much of the best software was designed to be content-free or ‘framework’ in nature and could therefore be applied in many subject areas through the provision of appropriate data files. These data files could be devised with teaching materials by third parties and disseminated through educational exchanges. The critical factor here was the open nature of the standards for the files enabling anyone interested to construct new learning examples.

The battle between proprietary and open standards has almost invariably been won by the latter. The government-sponsored BBC computer in Britain and other countries lost out to the PC platform. The X400 OSI standards for email addresses and directories lost out to the Internet Engineering Taskforce’s RFC 451 standards. Applebe (2003) predicts that Microsoft having adopted open XML for underlying file standards may shortly lose out in the competition with Linux and Open Office. Richard Stallman has argued since 1984 that the term ‘free software’ should be used and wrote the GPL (General Public Licence) to make the conditions clear. Linus Torvalds released the Linux operating system on 17th September 1991 under a modified version of the GPL (Rivlin, 2003, p.206). The ‘Blue File’ software pioneered this approach in the context of school education and achieved global impact.

4. OFFICE-CENTRIC VERSUS LIBRARIES OF CURRICULUM SOFTWARE

The nature of classroom computer use in more recent years bears comparison with the strategies embedded in the 'Blue File'. A seven-year longitudinal study of computer use in Tasmanian primary classrooms revealed a disturbingly monotonous classroom software diet. Using categories established by a national discussion process (Fluck, 1998) observations were made by over a hundred pre-service teachers each year to assess which of these categories were used by pupils in the classes they observed over at least a two week period. Publishing electronically or to paper was found in a steadily increasing proportion of classrooms. However, personal communications and problem solving remain restricted to a small fraction of these same classrooms (Table 2).

Table 2. Proportions of primary school classrooms where pupils were observed using computers in identified learning modes

Year	Publishing	Problem Solving	Independent Learning	Researching	Communicating
1995	75%	35%	36%	18%	6%
1996	78%	63%	44%	37%	11%
1997	82%	51%	52%	48%	16%
2000	86%	34%	59%	45%	35%
2001	85%	38%	50%	62%	42%
2002	86%	41%	66%	59%	39%
2003	94%	34%	72%	70%	42%

The results illustrate a trend to concentrate computer use in schools to the few basic software applications with which the teacher is familiar. This is often used to replicate activities that previously existed in the curriculum making ICT (where infrastructure is unreliable) a headache rather than an essential. This trend away from a rich repertoire of curriculum based software towards learning restricted by exclusive use of a basic 'Office' suite of productivity software can be seen as restrictive. It may satisfy a pedagogical rationale to prepare pupils for the world of work but does not easily translate into better learning in the wide range of subjects studied at school.

5. CONCLUSION

If ICT is to transform the curriculum then teachers need a rich library of software, either through open sources or from good collections of on-line

learning objects. Open source projects in Australia include BELTS (Basic E-Learning Tool Set) sponsored by the government as a free open source software project to trial the use of on-line learning objects, and Moodle, an educational course management system used extensively in Western Australia (Oliver, 2003). Both of these can be found in Sourceforge.net, which hosts over 65,000 collaborative software projects (Goetz, 2003, p 166). The open source method has been applied to other joint ventures involving volunteers. Project Gutenberg uses distributed proof-reading whereby many copy editors check copyright-free texts for accuracy prior to dissemination (Goetz, 2003, p 166). The open source on-line encyclopedia 'Wikipedia' was designed using open source web-design software called Wiki by Jimmy Wales and his team from 2001. It had 150,000 entries, and hits to its web-site surpassed those of the Encyclopedia Britannica in 2003 (Goetz, 2003, p 166). The government education department for the state of Victoria in Australia sponsored a competition for school students to submit open source educational software (Wide Open Source, 2003), demonstrating a commitment to spreading the concepts through the education of students.

The Learning Federation <<http://www.thelearningfederation.edu.au>> is a joint venture involving government agencies which is developing a range of on-line learning objects for Australia and New Zealand. It is not the current intention of the Learning Federation to release its learning objects as open source code, reflecting the high cost of production. It remains to be seen what value these objects have as a market becomes established for such content. Characteristics of such a market may include language and cultural localization, and it will be important to link each object with appropriate points in the customer's school curriculum. As on-line learning objects become more widely available, they demonstrate the importance of the 'Blue File' concept of framework software, where different data files are accessed to provide learning experiences for different subject areas. However, the web-browser and plug-ins used to access the learning objects do not appear to be as flexible in terms of input and output device range as the 'Blue File' software showed possible.

REFERENCES

- Applebe, W., 2003, The future of open source software, *J. of research and practice in IT*, 35:4. On-line at <http://www.acs.org.au/JRPITVolumes/JRPIT35/JRPIT35.4.227.pdf>.
- Department for Education and Employment, UK, 1998, *Summary of circular 4/98: Requirements for courses of initial teacher training*. Author: London, England. On-line at http://www.dfec.gov.uk/circulars/4_98/summary.htm on 5th June 2000.

- Fluck, A., 1998, The KITO's - Key information technology outcomes for school students. *Australian Conference on Computers in Education*: Adelaide, Aust. On-line at http://cegsa.sa.edu.au/conference/acec98/papers/p_fluck.html
- Goetz, T., 2003, Open source everywhere, *Wired*, November, pp. 158-208.
- HMSO, 1990, *Technology in the National Curriculum*, HMSO, UK: Department of Education and Science and the Welsh Office.
- Learning to Cope 1986/7*. Educational Computing Ltd/EMAP: London, England.
- Morgan, L., 1990, *Resource and Software Directory 1990*. L. Morgan: Melbourne, Australia.
- Oliver, Nick, 2003, *Open source software in education*. Canberra Schools Linux User group. On-line at <http://www.linux.act.edu.au/aeuopen/aeuopensource.html>
- Rivlin, Gary, 2003, Leader of the free world. *Wired*, November, 152-157 & 204-208.,
- Saunders, Peter, 1984, *Micros for handicapped users*. Helena Press: Whitby, England.
- Vincent, A. T., (Ed.), 1989, *New Technology, Disability and Special Educational Needs: some case studies*. Empathy Ltd c/o Hereward College of Further Education for the Open University and the Further Education Unit: Coventry, England.
- Wide Open Source, 2003, *Wide open code: The open source Victoria scholar challenge*. On-line at <http://www.osv.org.au/index.cgi?tid=6>