Open Source as Appropriate Technology for Global Education

Dr. Patrick Carmichael
Lecturer in IT Education, School of Education, University of Reading, UK.
p.carmichael@reading.ac.uk

Mr. Leslie Honour
Coordinator of Secondary IT, School of Education, University of Reading, UK.
l.honour@reading.ac.uk

Open Source as Appropriate Technology for Global Education

Abstract

Economic arguments for the adoption of 'open source' software in business have been widely discussed. In this paper we draw on personal experience in the UK, South Africa and Southeast Asia to forward compelling reasons why open source software should be considered as an appropriate and affordable alternative to the currently prevailing dependency on large commercial organisations and proprietary products in the field of education.

The dynamic and responsive nature of 'open source' software and the existence of freely available documentation and online communities offers an opportunity for educators, network administrators and software developers to participate in the development of resources appropriate to local needs while developing their own skills.

We identify a range of critical development tools such as Perl and Linux, alongside a more specific application, Basic Support for Cooperative Work, which has great versatility for customising to fulfil specific educational needs and for the development of collaborative on-line learning communities.

Introduction

Our interest in the development of open-source software and of the use of the internet for collaborative educational projects has its origin in our experiences as IT trainers and developers in a range of educational contexts in the United Kingdom, Southern Africa and South East Asia. Although working in different contexts and within different educational paradigms, our common experience is of schools, colleges and universities struggling to keep pace with rapid developments in IT. Opportunities for appropriate training are limited, with many educators being self-taught IT users. At the same time, rapid developments in computer hardware and the demands of current software means that educational establishments either have to work with technology which in other contexts would be considered obsolete, or have to redirect funds from other areas. Even if current hardware and software is obtained, either as a result of reallocation of funds, through community fundraising or from outside sources, institutions can then be 'locked' into the long-term use of specific proprietary products and find that they have added financial commitments: hardware and software upgrades, technical support, maintenance contracts and platform- or software-specific training. We would like to give a brief comparative snapshot of some of our recent experiences and observations.

Schools and colleges in the United Kingdom are currently involved in a number of government-funded initiatives: to provide up-to-date computers in schools, to train all teaching staff and to establish a national IT network (the ‘National Grid for Learning’). The experience of many educators, however, is of mixed-platform environments in which many of the computers are incapable of operating current software (McKinsey, 1997; Stevenson, 1997) and in which much of the access to the internet and to training is mediated by commercial concerns, often consortia, with little or no expertise in educational IT. There is little apparent notion of the potential to create on-line communities though which, for example, curriculum materials and teaching and learning methods, especially those concerned with subject specific applications of IT, could be created, tested, discussed and revised; a bottom-up, rather than a top-down modus operandi.

By comparison, we have encountered a similar, although greatly exacerbated, set of problems relating to the successful integration of IT into the education systems and curriculum of developing countries where we have worked as consultants. A group of secondary Schools in Gauteng Province, South Africa, for example, having been equipped with networked computer suites as part of a Schools-University Partnership project, found it difficult to expand their capabilities through operating system upgrades and internet connections without also upgrading hardware. This then necessitated upgrading the commercial server software, and project managers were faced with dilemmas over how to fund both IT developments and training needs of staff, both of which were handled by commercial concerns. Similarly, government sponsored participants from eight countries of South East Asia on an intensive training course in the development of interactive multimedia resources for web-based technical and vocational training, described a common range of constraints, both technical and organisational, which they would face in trying to implement the use of these resources in their respective countries: lack of present generation hardware and software with the constant pressure to upgrade, lack of national network coverage, low bandwidth, high cost and sometimes unreliable communications networks and a lack of both systems management and end user skills.

In spite of the reservations which these cases might imply, we remain advocates of IT as a key resource in the development of educational provision world-wide, and are currently involved in a number of projects addressing initial teacher education, teachers' continuing professional development and the effective dissemination of classroom resources. What underlies all of these projects is our commitment to avoiding the 'proprietary lock-in' described above; to training coupled with provision; and to the idea of creating an enduring networked community of IT users and developers, within which support, training and peer-review can occur. Our adoption of open-source software is an extension of what we regard as best practice in educational research and development.

Open Source software and the Internet

'Open Source' software is the current manifestation of a culture of collaboration which has existed since the early days of computer science and having its roots in what Levy (1984) calls the 'hacker ethic' exemplified by the computer science fraternity at the Massachusetts Institute of Technology from the 1960's onwards. Levy's account concludes with a characterisation of Richard Stallman, guiding force behind the 'Free Software Federation', as 'the last of the true hackers' working in virtual isolation to develop and promote 'free' software at a time of increasing commercialisation. Levy's prognosis was, however, overly pessimistic. The ideal of hardware and software development occurring within a cooperative network was to survive in a number of contexts, and was simultaneously a moving force behind the development of the internet and the World Wide Web, and a beneficiary of the virtual community of IT users and developers which it connected.

Expanding from 870 publicly-accessible 'sites' in November 1994 to over 5 million distinct sites by April 1999 (Stein, 1999) the internet represents the fastest-growing mode of communication currently available. Of the 5 million sites reported by Stein (current estimates as of September 1999 run at more than 7 million), some are tiny online documents such as 'personal home pages' while others represent entire educational institutions, libraries databases, corporate e-commerce sites or large sites such as 'Geocities' which hosts web pages for an estimated 1.7 million individual users. Distributed across the internet are a large number of what Rheingold (1984) has called 'virtual communities', amongst which are numbered the various groups of open-source developers and users. Additionally, there are explicitly educational 'communities' within which teachers and students share resources, review work and work collaboratively on, for example, writing and programming projects (Bruckman 1998). Mansell and Wehn (1998) have discussed strategies for building "knowledge societies" and the potential uses of ICT for sustainable development, while Nardi (1999) has referred to "information ecologies" as "a system of people, practices, values and technologies in a particular local environment…..In information ecologies the spotlight is not on technology, but on human activities that are served by technology." It is this feature which can be considered to make a technology 'appropriate'.

While the prevailing perception of the development of the internet and specifically the World Wide Web may be one of being built and maintained using proprietary products , it has been established that, across the internet as a whole, the majority of web servers operate Apache web server software. This web server is increasingly likely to be running on one of the free Unix variants (FreeBSD or Linux); the majority of users' email is handled by Sendmail, and developers and administrators are likely to use one or more of a set of scripting languages (Tool Command Language or Tcl, Perl and Python) to monitor and regulate access to resources and to create web-accessible databases and applications. Web pages are commonly constructed automatically using Perl (and to a lesser extent Python) which also power many of the well-known internet search engines and catalogues.   It must be stressed that open source does not represent a solution inferior to proprietary products, or that it is only suitable for operating in the low-traffic backwaters of the internet: many of what Stein (1999) calls the 'massive landmarks' of cyberspace, such as the Yahoo! catalogue and search engine, are dependent on a range of open source software. What all of these pieces of software - operating systems, servers, programming environments and applications have in common - is a commitment to making their source code public and to continuous elaboration by the user and developer community.

Apache (regarded as one of the most effective 'open source' projects to date) currently has a 55% share of the world server market (according to the Netcraft survey for September 1999) despite, as Raymond (1999) observes, there being no legal owner, little promotion and no 'service contracts', just a group of IT professionals who pool code, operate a peer-review system and provide information and upgrades to colleagues and clients.   This pattern of collaboration and the resulting flexibility of the software produced are the characteristic features of open source software and it is these feature which we will subsequently argue make it an appropriate basis for the development of educational IT.

The Open Source Summit and the Open Source Definition

In April 1998, leading producers of 'free' software including Apache, Linux, Tcl, Sendmail and others including Netscape, who had recently released the source code for their 'Navigator' web browser, held a 'summit' to discuss the development for a common strategy for promotion and further development of their products. Guido van Rossum, creator of the Python object-oriented scripting language identifies a key issue is his account of the meeting (1998): "[While] a large number of developers said their initial or ulterior motivation [for making source available] was moral/ethical: they believe that it is "the right thing to do" ... almost everyone present was involved in an attempt to commercialise their software ... without making their sources proprietary". Van Rossum continues: "everybody is working on a sustainable business model that produces a sufficient revenue stream to pay for developers and a support organisation without giving up the advantages of open-source software". What emerged from the 'Open Source Summit' was an 'Open Source Definition' developed by Eric Raymond which includes the following details:

> The program must include source code, and must allow distribution in source code as well as compiled form. Where some form of a product is not distributed with source code, there must be a well-publicised means of obtaining the source code for no more than a reasonable reproduction cost -- preferably, downloading via the Internet without charge. The source code must be the preferred form in which a programmer would modify the program ...The license must allow modifications and derived works, and must allow them to be distributed under the same terms as the license of the original software ... the license must not restrict anyone from making use of the program in a specific field of endeavour.

This Open Source Definition, discussed along with other software licences by Perens (1999) has been scrutinised closely and has been subject to some criticism. Guido van Rossum has yet to apply the label to Python, although it is fully 'open source'. Richard Stallman has mounted a spirited defence of the notion of 'free software' (1999), and is suspicious of the commercial orientation of many of those now espousing the Open Source Definition. Raymond's response to the question of how a reliable revenue-stream might be generated by an 'open source' enterprise while maintaining the open source 'ethic' is contained in a recent paper, 'The Magic Cauldron'. Raymond is best known for his earlier paper 'The Cathedral and the Bazaar' in which he proposed that a 'bazaar' style of software development, characterised by decentralisation and cooperation, could lead to increased productivity, reliability and quality and cited the Linux operating system as a case in point. In 'The Magic Cauldron', he explicitly abandons the notion of 'gift culture' inherent in his earlier model and describes a range of business models for software developers based on the experience of Red Hat Linux, the most 'commercial' of the Linux operating system distributions (which he characterises as 'selling the recipe and opening a restaurant') Apache (cost-sharing), Netscape (loss-leading and maintenance of market position), O'Reilly publications (documentation) and on assessments of how companies such as Sun or America Online might embrace the Open Source definition in the future.

Basic Support for Cooperative Work (BSCW): an open source project application with wide potential in education

The open-source programming languages and resources we have described are characterised by their 'scale-neutrality'. Open source 'projects' range from what Perl programmers refer to as 'one-liners' such as sorting lists, through medium-sized CGI (Common Gateway Interface) applications which handle form data from the WWW and generate customised web pages for users, to very substantial projects, of which the Linux operating system and its graphical interfaces represent the best known examples.

Another example of a large and developing project, written in this case almost entirely in Python, is Basic Support for Cooperative Work (BSCW) , a software project developed by FIT, the Institute for Applied Information Technology, a research unit of GMD, Germany's national research center for information technology, and partially funded by the European Union through the CoopWWW project and the CESAR project of the EU's Telematics Applications Programme. Its designers describe it in the following way:

> The BSCW system supports collaboration by providing shared workspaces over the

> Internet. A shared workspace allows storage and retrieval of documents and sharing information within a group. This functionality is integrated with an event mechanism to provide each user with an awareness of the activities of others within the workspace. It comprises numerous features, e.g., support for threaded discussions, version management of documents, group management, search features and many more. The system is designed primarily to support self-organising groups. (BSCW, 1999)

The BSCW system (fully described in Bentley et al, 1997) addresses the need for flexible access to online resources without the need for specialist client software, since the entire collaborative workspace is accessible through a standard internet browser such as Microsoft Internet Explorer or Netscape Navigator and, in our experience, it is possible, though less easy, to gain access and navigate a BSCW workspace using the Lynx text-only browser which makes very limited demands on the client computer. It is also tailored for asynchronous communication - documents and contributions - can be uploaded to a workspace either for public or limited distribution and remain in place. On subsequent 'visits' the workspace, the originator can review any access to, comments on, or revisions (if permitted) of their work. Access controls allow parts of a workspace to be configured in different ways - for anonymous file transfer (FTP), for secure website management, for email and conferencing, or for document management and version control. Links to other internet resources, results of internet searches, and documents in a wide range of formats including text, graphics and spreadsheet data can be uploaded, viewed and converted between formats in order to ensure backwards and cross-platform compatibility. In addition, advanced features such as video-conferencing may be integrated into the conferencing facilities. All of these functions have considerable value in a range of educational contexts, and the BSCW system as a whole has all the elements necessary for the support of a substantial and participatory online community-based development project.

Furthermore, the transition from 'user' to 'developer', made possible by the open source approach is supported in that GMD maintain a public server where workgroups can set up workspaces. Once appropriate levels of expertise have been reached and a suitable host server located, the entire set of workspace software can be downloaded and administration is handled locally under a free licence (available to educational organisations) supported by comprehensive documentation, a help system and an online developer community. In our experience, training users to manage a workspace is reasonably straightforward. For our student teachers and their respective partner school mentors, we have recently created collaborative on-line discussion groups in the hope that these will act as a catalyst of change for the school departments in which they are working. Complementary to this would be the training of IT managers to a level where they would be in a position to administer their own BSCW server. However, this is a much more serious undertaking; such building of local capacity would offer a wide range of opportunities to develop online resources and to maintain the status of projects in the developer community.

Conclusion

Open source developers are currently being challenged to maintain the distinctive features of open source software within sustainable business models and, as Dougherty (1999) states, they may find the transition from dealing with 'users' to 'customers' problematic. In the education world, however, there already exists a long-established legacy of shared responsibility for projects; 'distributed development'; peer review; and regular revision in the light of experience. These are all characteristic practices of open-source development; hence, it could be argued, that open source would find a natural home in education. We would encourage developers to consider the alternatives offered by working closely with governmental and non-governmental agencies, including but not exclusively within the education sector, where the flexibility, community-orientation and capacity-building potential of open source software have a wide range of applications and enormous advantages. We acknowledge that systems management training, including Linux and Perl in particular, would be fundamental to such a strategy. However, education has generally failed so far to provide any form of training at this level to IT Co-ordinators. If it were to be based upon open source, we believe that the tangible long term benefits would far outweigh the initial investment costs and that most of the current problems identified, especially the 'proprietary lock-in' issue, would gradually wane. Pilot surveys and conferences which we have conducted amongst school and college co-ordinators in the UK have indicated an immense enthusiasm for BSCW in particular and open source more generally. We are embarking upon a range of research projects, initially based upon collaborative curriculum work between our own university and some of the partner schools of our trainee teachers. We hope to extend this to developing countries, where an open source strategy would have relatively greater potential benefits, in the near future.

Glossary and Sources of Information

Apache: HTTP Web server application. Information and downloads from the Apache Software Foundation at http://www.apache.org.

CGI (Common Gateway Interface): Used as a standard for the processing of data gathered from web pages - surveys, bookings, feedback forms and automatic subscription and mailing systems. It also permits the generation of 'on the fly' webpages which may contain user-specific information. Judging from the contents of the CGI 'clearing-house' at http://www.cgi-resources.com, the vast majority of CGI scripts are written in Perl.

Linux: Originally 'reverse-engineered' by its creator, Linus Torvalds, as a programming project, Linux is now, thanks to its large developer community, the fastest-growing variety of the UNIX operating system. While the entire operating system can be downloaded from a number of internet sites, 'commercial' distributions are also now available including not only the Linux Kernel but a range of other open-source applications and interfaces. There are a great many websites devoted to aspects of Linux development, but http://www.linux.org is a good starting point.

Perl (Practical Extraction and Report Language): Versatile programming environment characterised by close integration with the system on which it is running, an ablity to act as a 'glue' language linking other applications, a powerful set of regular expression tools (which allow searching, matching and substitution of text), and a vast collection of add-on 'modules' available from CPAN (the Comprehensive Perl Archive Network). CPAN, together with platform-specific information and downloads are available at http://www.perl.com

Python: Flexible object-oriented scripting language with a notably 'clean' and consistent syntax. Its advocates cite its ease of use and its integration with C, Java and the Windows COM API. http://www.python.org

Sendmail: While users may have a variety of email client software on their own computers, Sendmail is a server-based email processor used by corporations, educational institutions and Internet Service Providers. The Sendmail consortium is at http://www.sendmail.org

Tcl (Tool Command language): Specifically developed as a scripting language used to 'glue' other applications together and which could be embedded in other programs. Its companion Tk (Tool Kit) extension allows the construction of graphical user interfaces which replicate the 'look and feel' of X-Windows, Windows 95/98 and Macintosh applications. http://www.scriptics.com

References

Bentley, R., Appelt, W., Busbach. U., Hinrichs, E., Kerr, D., Sikkel, S., Trevor, J. and Woetzel, G.(1997), 'Basic Support for Cooperative Work on the World Wide Web' *International Journal of Human-Computer Studies*, 46(6): Special issue on Innovative Applications of the World Wide Web: 827-846.

Bruckman, A. (1998) 'Community Support for Constructionist Learning' *Computer Supported Cooperative Work* 7:47-86

BSCW (nd) *BSCW Overview* http://bscw.gmd.de/

Dougherty, D. (1999) 'The Selling of Open Source' *Web Techniques* 4(9): 88

Levy, S. (1984) *Hackers: Heroes of the Computer Revolution* Dell Publishing Group. Reprinted by Penguin, Harmondsworth, 1994 with a postscript by the author.

Mansell R. and Wehn, U. (eds.) (1998) *Knowledge Societies: Information technology for sustainable development* , Oxford University Press

McKinsey & Company (1997) *The future of information technology in UK schools (The McKinsey Report),* McKinsey & Company, London.

Nardi, B.A. (1999) *Information Ecologies: Using technology with heart*, MIT, Cambridge, Massachusetts

Netcraft (nd) *The Netcraft Web Server Survey* http://www.netcraft.com/survey/

Perens, B. (1999) 'The Open Source Definition' in: *Open Sources: Voices from the Open-Source Revolution* O'Reilly Publications , Sebastopol CA: 171-188.

Raymond, E. (1998) 'The Cathedral and the Bazaar' http://www/tuxedo.org/~esr/writings/ cathedral-bazaar.

Raymond, E. (1999) 'The Magic Cauldron' http://www/tuxedo.org/~esr/writings/ magic-cauldron.

Raymond, E. (1999) 'The Open Source Definition v. 1.7' http://opensource.org/osd.html.

Rheingold, H. (1984*) The Virtual Community* Martin Secker and Warburg,

Stallman, R. (1999) ' in: *Open Sources: Voices from the Open-Source Revolution* O'Reilly Publications, Sebastopol, CA: 53-70.

Stein, L. (1999) 'How Shall I Measure Thee? Let Me Count the Ways' *Web Techniques* 4(7): 16-19.

The Independent ICT in Schools Commission (1997) *Information and Communications Technology in UK Schools - An Independent Inquiry (The Stevenson Report)*, SRU Ltd., London.

Van Rossum, G. (1998) 'Open Source Summit Trip Report' http://www.linuxgazette.com/issue28/rossum.html