The Free Future of General Medical Software

Michael Bonert, MASc (Biomed Eng), BASc (Mech Eng), MD Candidate - Class of 2007
Bradley MacIntosh, MSc, BSc

Introduction

Using computers can be a frustrating experience. Anyone who has used a computer may have found themselves, at one time or another, cursing in frustration. Why doesn't the computer do what it is supposed to? Why doesn't this program work? With these questions in mind, we set out to understand the role of computers in the field of medicine, and the future directions of computing.

The first section of this article discusses medical software. The later part examines general trends in software and relates them to developments in medical software. For the purposes of this article, "medical software" is defined as software used for hospital administration, radiology archiving, and to run a family doctor's office. We surmise general medical software will be quite different in the future.

Part I
Medical Software

What is medical software?

One definition is: a computer program that is used to organize and communicate health information. Many experts agree software will play an increased role in patient care. Software is useful for tracking and coordinating medical information. For example, it can remind a physician to give a patient their tetanus booster shot or check a patient's blood pressure and cholesterol.

In this context, one may ask the question: why is this important to me as a physician or trainee in medicine? The answer to this question consists of these two important points: (1) software will be part of the next leap forward in the quality of care and (2) it could become a significant health system cost-driver. Software might become a major financial burden to the health system because software vendors which withhold the source code (the 'blueprints' of software) from their customers, which we will refer to as "lock-in software providers," would like to double health system information technology (HSIT) revenues.[1]

Software as the Next Leap Forward in the Quality of Care

Drug reactions are a leading cause of morbidity and mortality.[2] Software's role in tracking prescription drugs is worthwhile from a safety point of view.[2] Furthermore, using software for electronic records allows us all to do better epidemiology and help automate easy calculations, such as BMI (body mass index). Beyond these obvious examples, most decisions in medicine follow a logical sequence that computers can guide. They can also help us understand decisions and help standardize treatments. It makes sense to have electronic antenatal forms. Further, many medical procedures vary significantly based on geography alone.[4,5] Having software that keeps track of decisions and how decisions are made can help standardize procedures and allow people to analyze and rewrite the software to help guide decisions and improve care. Decision support software is already in use at Telehealth Ontario[6,7] and in some family physicians' offices.[8,9] This is good news for doctors, since many of them would prefer to deal with patients as opposed to complicated decision trees that are part of today's immensely complex treatment algorithms.

The Relevance of Understanding Software

As the role of software increases, so too will the consequence of malfunctioning software. For instance, a software error in a drug dispensing pump caused significant safety problems for patients.[10-12] A software bug in the Canadian-made Therac-25 radiation machine caused at least six separate accidents and led to the death of three patients.[13] Since poorly written software can have serious consequences, it is natural to ask about how the software works and how we can ensure that potential harm is minimized.

Software as a Significant Health System Cost Driver

In the US, annual health system information technology (HSIT) expenditures were estimated to be more than $20 billion US ($24.8 billion Cdn).[14] In the UK in 2000 health IT spending was estimated to exceed £1 billion ($2.3 billion Cdn).[15] Canadian health system IT expenditures are approximately $2 billion Cdn.[1] In the interest of containing health costs and improving our public health system, it is worth asking whether we need to pay more for a more comprehensive health system IT.

Part II
General Software and Why 'Free' Works

A Shake-Up in the Software World is Coming

With the success of the Internet, new ways to develop software have emerged which are reminiscent of early computing days.

In the 1970s it was customary to share source code (the 'blueprints' of software). This breed of software development is now re-emerging on a larger scale resulting in software that is based on open standards and is free (free as in "free speech," as opposed to free as in "free beer"[16]) or open source.[17] This way of producing source code is changing the fundamental dynamics of software development. "Free software" is not (necessarily) free as in price. However, it is almost generally less expensive than lock-in software because it generally has (1) lower software licensing costs and (2) there is more competition on service.[18-20] It has lead to increased standardization, a requirement for general office computing and medical software in order to be useful. An additional benefit is that the software delivered has been low in cost and of high quality.

Open Source Software and Free Software differ somewhat, but are often used synonymously. The difference between these two terms is discussed by Richard Stallman, founder of the free software movement.[21] We will collectively refer to them as FOSS (Free and Open Source Software).

General Software is a Service

General process/business/management software, as pointed out by Open Source programmer/historian/anthropologist Eric Raymond, is really a service industry, not a manufactured good.[22] Stores that sell software know this; if a software vendor goes bankrupt, or discontinues a product line the value of the latest release (and prior releases) is significantly reduced. This is because a general process/business/management software's value is really based on a future release, on its ability to read common data formats, and on service. This is unlike a manufactured good; an automobile has value even if the manufacturer goes bankrupt or a new model has been released.

Software sold as a manufactured good also has other undesirable consequences: it leads to insecure, low quality software. A company that makes its money by selling bits or boxes (of software) will see support/service as a cost and is happiest if you just put the software on the shelf (aka "shelfware") or use a very small set of the features of that software. This is often the opposite of what a company that sells service wants, namely a functional product that works perfectly (hence generating fewer service calls).

The Importance of Inter-Operation for General Software

Inter-operation of medical software is key for usability because it allows people to share information without costly and time-consuming transcription and/or format conversion. If prescriptions are going to be done electronically in the future and ordered via computer, the pharmacist's computer has to understand the physician's computer. This can be accomplished with open standards and history shows us that true standards encourage adoption. The reality is that people avoid the toll road if there is a free alternative that gets you to the same destination in just about the same amount of time. As an illustration of this idea, consider the open standard Transmission Control Protocol/Internet Protocol (TCP/IP), the communication method of the Internet, which completely displaced Novell's proprietary IPX protocol and made it irrelevant.

Lock-in software vendors do not want inter-operation. In fact, open standards are hurtful to the bottom line in companies that depend on lock-in because they commoditize protocols and make sharing information between competing software programs easier. Microsoft documents show that their strategy has been to break open standards and de-commoditize them where ever possible with extensions they claim have value,[23] something that is illustrated by the battle between Sun Microsystems and Microsoft over Java.[24] Lock-in software companies want a captive market so they do not lose their customer base and can easily dictate prices and force upgrades and thereby increase profits. This model of software development imparts power to the software vendor. It has made some people fabulously wealthy, but because of a lack of competition, has provided software users a good deal of inferior software that is insecure, unstable and incorporates schemes that limit user control, such as digital rights (or perhaps more appropriately "restrictions") management.

Increasing FOSS based HSIT (health system information technology) in Canada can help remedy many of the problems seen today. The result will be more competition between software companies in the customization of the (free and open source) software and the support for it. Adopting this framework will expand the commons that is shared globally (the source code), yet still leave plenty of room for free market innovation and involvement. In the computer industry as a whole, a number of companies are currently pursuing this strategy and making their money off of service and/or hardware. The largest is IBM, which has invested billions in FOSS (Free & Open Source Software)[25] and promotes Linux, the most visible Open Source Software project. Smaller companies, like Redhat, Suse – now owned by Novell) are viable businesses and rely, to a large part, on support and customer customization requests for revenue. Business strategies for open source companies are described by Bruce Perens, a leader in the FOSS community and the author of a widely used definition of free software.[26]

The Danger of Lock-in Software

The bane of relying on proprietary software is that one never knows how long it will be supported. The investment of time in learning an interface can be lost if the company forces upgrades or goes bankrupt and one is forced to switch to software from another vendor. By contrast, users of established FOSS will, generally, only ever have to learn one interface. Regardless of hardware changes and operating system changes an interface can be maintained if one has the source code, and changes to the software happen merely behind the scene.[27]

Why does free software work?

Naturally, physicians are interested in software that works effectively and efficiently. Projects that are not mature evolve over time as people write custom extensions or enhancements to do important tasks such as fecal occult blood testing in elderly patients. Once an enhancement is written, the authors of that extension are encouraged by economic incentive to submit it to the project leaders/source code maintainers. By doing this they save future work, as they would otherwise have to maintain and update the enhancement and adapt it as the software increases in functionality, from changes made by other people.[22]

The vast amount of free code base in the world means FOSS minimizes the costly duplication of effort. How much free code is out there? In 2001 Red Hat Linux was over 30 million lines of code. If it were commercially developed in the USA it is estimated that it would have cost over $1.24 billion ($1 billion US).[28] By comparison, Windows 2000 in 2001 was estimated to have approximately 35 million lines of source code.[29]

Another reason FOSS is effective is because of the speed in which software bugs are detected and subsequently fixed. Often software bugs are shallow and small. In essence, FOSS development takes advantage of informal peer-review.[30] When lock-in software companies lock up their source code, they lose the possibility of having others make improvements. Proof of this comes from the open kernel (Linux), which has significantly lower number of bugs than found in lock-in (commercial) software.[31,32]

The Complexity of General Medical Software

Experience has shown that the information systems in health systems are difficult to commission, evaluate and purchase.[14,15] Health systems are very complex, so development of health system software must occur by an iterative process. Lock-in software in this environment is a costly proposition as many large software projects typically require significant changes after being written. At this stage lock-in software vendors have the buyers at their mercy because the buyer cannot make the changes themselves and only has the choice of paying up or walking away from the project and starting over (and paying for transition costs to a new system).[15] Many large software projects have ended in failure because they did not meet their objectives and modification was seen as too costly. As a result, the source code (the blueprints of software) of many lock-in programming efforts disappear.[14,33,34]

Health as a Shared Resource and Leveraging Foreign Expertise

With globalization, health has become more international and promoting good health has become a transnational endeavor. Disease does not respect national boundaries. So, we must find ways to share health information and extend health technology to the developing world.[35] Canadians have taken the lead in promoting the spread of health information; the Canadian Medical Association Journal (CMAJ) believes medical information should be freely available to those who benefit from it.[36] Increased adoption of FOSS would promote technology transfer and expand the global health information commons, since FOSS is transnational. Denying poorer countries access to medical software only adds to the health care divide caused by the barriers these countries face when seeking patented pharmaceutical treatments.[37] Medical software should not become another source of inequality. Further, unlike much of medical research, software development is not a resource intensive endeavor. Also, after development costs, software has an almost zero marginal cost to reproduce/or copy. Thus, further adoption of FOSS promotes health around the globe, makes health technology more accessible and allows people with fewer resources to make meaningful contributions to our health information infrastructure. Stated differently, adoption of FOSS in Canada allows Canadians to leverage foreign software expertise.

Outside of North America the movement toward FOSS is strong. Almost all of South America is moving toward adopting FOSS.[38] Peru has made a very convincing case that governments should be using FOSS[32] and is widely expected to soon pass a law that mandates open source software for all government business. Brazil, having received the ultimate blessing by the Massachusetts Institute of Technology[39,40] is expected to use FOSS for a government program that will sell computers to one million poor.

The Success of Free Software

The success of free software in general shows where we are going.[41] Some of the most visible companies and institutions run GNU/Linux (also known simply as Linux), such as Google, Amazon.com and the White House. Linux is on handhelds, appliances, robots, NASA supercomputers and in cameras.[42] Over 20 different cell phones are being developed that make use of Linux. IBM has invested billions in Linux.[43] Oracle, the de-facto enterprise database, has been ported to Linux and the CEO of Oracle has predicted that proprietary operating systems (Unix, Windows) will be wiped out of the data centre by Linux.[41,44] Open source Apache is the most popular web server program on the web.[29] It has over three times the market share when compared to Microsoft[45] and current trends suggest Apache's dominance will continue. On the desktop, GNU/Linux has now displaced Apple from the No. 2 spot.[46]

Interest in the European Union in OpenOffice (a free alternative to MS-Office) is increasing, the city of Munich has decided to adopt it as a standard and will migrate all its desktops away from MS-Office[47] and the Windows operating system. Government bodies in Texas and Israel are also moving toward OpenOffice.[48] Further, there are formal discussions about making the freely available OpenOffice document specification into an ISO standard.[49,50]

In medical software there is also a trend toward FOSS. A number of free medical software packages exist and are improving each year. Five years ago, a BMJ editorial predicted that the future of medical software will be free.[15] Today, a dominant hospital management software known as VISTA (Veterans Health Information Systems and Technology Architecture) and used worldwide is free.[51] There is OSCAR McMaster,[52] which is free family practice management software that was developed by a U of T alumnus. Also, there is a free PACS system.[53] Many other medical FOSS

projects are in development and the website EU Spirit[54] has a large list. It is not an accident that the Australian General Practice Computing Group has come to the conclusion that open standards are important and free and open source software best serves the interests of physicians and their patients.[55]

The CMA's Purchase of HealthCare Software Inc.

The Canadian Medical Association (CMA) recently bought a lock-in software vendor (HealthCare Software Inc.) for an undisclosed amount.[56] With this purchase there is an opportunity for Canadian physicians to take the lead in HSIT.

If the CMA's software vendor were to adopt a free software license (such as the GPL) Canadian physicians could:

(1) Leverage the knowledge of foreign computer experts to develop the best (family practice) HSIT.
(2) Be assured that their own professional association will not overcharge them for HSIT.
(3) Promote international health and the secure exchange of medical information.
(4) Build on and borrow from health system FOSS such as GNUmed, OSCAR McMaster, Drugref.org, and VISTA.
(5) Create software that is operating system independent and would run on all major operating systems (Windows, GNU/Linux, Mac OS-X).
(6) Increase adoption of HSIT and improve the health of Canadians.

The CMA could cover the cost of administrating the program by turning HealthCare Software Inc. into a services company that would sell services in software customization, general HSIT consulting and provide user support to physicians.

The CMA's alternative is pursuing a lock-in model of software development, which is less than ideal. The HSIT market place is currently crowded with small lock-in vendors, the software is relatively expensive (due to lock-in) and inter-operation is challenging (due to the large diversity of software). If the CMA fails to recognize the emerging software market place, it is certain that businesses will emerge that are based on FOSS (such as OSCAR McMaster), which leverage knowledge of experts worldwide, and is outside of the purview of the CMA.

Paying Twice for the Same Software

Carnall describes the situation in the UK:
"It is a customary Anglo-Saxon habit to laugh at the Academie Francais [sic] as it attempts to legislate for the language--yet we seem to be happy to allow our own institutions to do the same for our technical language--and charge us twice for the privilege: once from taxation as the system is developed, and a second time as publicly funded organizations buy licenses".[27]

In Canada, the situation is no different for software that is developed by our hospitals and governments. For instance, eFilm is medical imaging software that was developed at the University Health Network and is now being licensed back to Canadian hospitals by an American company, Merge eFilm of Milwaukee Wisconsin.[57]

When it comes to software, our American neighbours have a better idea. US government work is released to the public without copyright;[58] this is why we have VISTA (a dominant hospital management software package), which was developed by the United States Department of Veterans Affairs. The French government has gone a step further; their eGovernment initiative has released code under the GNU General Public License (GPL)[59] (the same license that GNU and Linux use), which ensures that "derivative works" (works created from previously copyrighted work)[60] of that public project remain FOSS and ensure that companies cannot create lock-in software out of work paid by the French taxpayer.

Looking Forward

Movement towards FOSS means that medical lock-in software vendors are likely to respond defensively. But it is not just software vendors who will defend the software industry status-quo. Personal communication with the editor of the Ontario Medical Review made it clear that, while FOSS arguments are sound in a medical context, there will be a challenge to implement them.[61]

France[62,63] and Germany[64] have committed to increasing the use of open source solutions. In the past, the German Federal Ministry of Economics and Technology supported the GNU Privacy Guard, a project developing encryption software using a FOSS model and licensed under the GPL. France believes it can achieve significant cost savings through the use of open source software.[62] It is likely that Germany, France or another nation in Europe will adopt the FOSS approach some time soon in the near future specifically for HSIT and contribute resources, source code and expertise that would be available to all parties that accept a FOSS development process. This would make using FOSS HSIT even more cost-competitive than it already is. Regardless, it is clear that adopting an FOSS approach and using FOSS already available for health system has the potential for significant cost savings.

The next few years will be years of transition. Free software will be installed concurrently with lock-in software. More universities will supply free software at cost and, like York University, more universities will host FOSS. More people will discover FOSS and become convinced of its value by projects like "The OpenCD" (http://www.theopencd.org/). Web pages will increasingly be designed to open standards as there is a shift away from insecure lock-in software[65] which the US government agency responsible for computer security has suggested is good to avoid.[66]

Conclusions

It is apparent that the free/open development model for software has lead to the development of very successful software, and that FOSS software has a number of advantages over lock-in software, since lock-in software source code is a secret and unavailable for peer-review.

FOSS should play a greater role in our health system. Developing technology in the public domain and then paying for it as a lock-in product is a common theme in our society. However, this scenario can be easily avoided in the area of medical software. There are both health and economic incentives to sharing health information technology. Further, doing so does not preclude free market participation in health information technology. It only re-

defines how technology providers compete. With this in mind, when will Canadian physicians and health IT experts embrace FOSS and use it to improve our health system?

Conflicts

The authors MB and BM are GNU/Linux enthusiasts. MB has made contributions to Mandrake Linux (now called Mandriva Linux) and the GNUmed medical software project and suggested bug fixes and/or improvements for wxWindows, Debian GNU/Linux and OSCAR McMaster. No form of financial compensation was involved in any of the above interactions.

Acknowledgments

We would like to thank Dr. Jim Busser and Dr. William Klein for their comments and suggestions.

References

1. CHITTA Retrieved April 21, 2005 from the World Wide Web: http://www.chitta.ca/about_chitta.html.
2. Baker GR, *et al.* The Canadian Adverse Events Study: the incidence of adverse events among hospital patients in Canada. *CMAJ*, 2004. 170(11): p. 1678-86.
3. Tamblyn R, *et al.* The medical office of the 21st century (MOXXI): effectiveness of computerized decision-making support in reducing inappropriate prescribing in primary care. *CMAJ*, 2003. 169(6): p. 549-56.
4. Kennedy J, *et al.* Variations in rates of appropriate and inappropriate carotid endarterectomy for stroke prevention in 4 Canadian provinces. *CMAJ*, 2004. 171(5): p. 455-9.
5. McIsaac WJ, *et al.* Otolaryngologists' perceptions of the indications for tympanostomy tube insertion in children. *CMAJ*, 2000. 162(9): p. 1285-8.
6. Ontario T. Available from: http://peel.cioc.ca/details.asp?RSN=15789&Number=14.
7. MOHLTC Retrieved April 21, 2005 from the World Wide Web: http://www.health.gov.on.ca/english/public/pub/ministry_reports/telehealth/telehealth.html. 1999.
8. Rachlis M, Prescription for excellence: how innovation is saving Canada's health care system. 2004.
9. Clinicare. Available from: http://www.clinicare.com/index.html.
10. Doyle J, Programming errors from patient-controlled analgesia. *Canadian Journal of Anesthesia*, 2003. 50: p. 855-856.
11. White E, A problem with the Graseby 3300 PCA pump. *Anaesthesia*, 1993. 48(11): p. 1013-4.
12. Ma M, *et al*., A potential PCA hazard. *Anaesthesia*, 1998. 53(3): p. 314.
13. Leveson. An Investigation of the Therac-25 Accidents, in IEEE Computer. 1993. p. 18-41.
14. Dorenfest S. Retrieved April 25, 2005 from the World Wide Web: http://www.healthcare-informatics.com/issues/2000/08_00/dorenfest.htm. 2000.
15. Carnall D, Medical software's free future. BMJ, 2000. 321: p. 976.
16. FreeSoftwareFoundation. Retrieved April 21, 2005 from the World Wide Web: http://www.fsf.org/.
17. OpenSource. Retrieved April 21, 2005 from the World Wide Web: http://www.opensource.org/.
18. Wheeler D, Why Open Source Software / Free Software (OSS/FS, FLOSS, or FOSS)? Look at the Numbers. Retrieved April 21, 2005 from the World Wide Web: http://www.dwheeler.com/oss_fs_why.html#tco. 2005.
19. Greene TC, Ballmer 'fesses up to Linux/Windows cost FUD. Retrieved April 21, 2005 from the World Wide Web: http://www.theregister.co.uk/2002/07/16/ballmer_fesses_up_to_linux/. 2002.
20. Cirillo R, Ballmer: Linux Changed Our Game. Retrieved April 21, 2005 from the World Wide Web: http://www.varbusiness.com. 2002.
21. Stallman. Retrieved April 21, 2005 from the World Wide Web: http://www.gnu.org/philosophy/free-software-for-freedom.html.
22. Raymond E, The Cathedral and the Bazaar. Retrieved April 21, 2005 from the World Wide Web: http://www.catb.org/~esr/writings/cathedral-bazaar/cathedral-bazaar/. 2000.
23. OpenSource. Retrieved April 21, 2005 from the World Wide Web: http://www.opensource.org/halloween/halloween1.php.
24. McCullagh D, Shankland S, Court curbs Microsoft Java distribution. Retrieved April 21, 2005 from the World Wide Web: http://news.com.com/2100-1007_3-1021452.html. 2003.
25. IBM, IBM bets 2.5 Billion on Linux for semiconductor manufacture. Retrieved April 21, 2005 from the World Wide Web: http://www-1.ibm.com/linux/news/semiconductor.shtml. 2003.
26. Perens B, The Emerging Economic Paradigm of Open Source. Retrieved April 21, 2005 from the World Wide Web: http://perens.com/Articles/Economic.html. 2005.
27. Carnall D. Retrieved April 21, 2005 from the World Wide Web: Open Source Software in Healthcare http://www.informatics-review.com/open.html. 2000.
28. Wheeler D, More Than a Gigabuck: Estimating GNU/Linux's Size. Retrieved April 21, 2005 from the World Wide Web: http://www.dwheeler.com/sloc/redhat71-v1/redhat71sloc.html. 2002.
29. Wheeler D, Counting Source Lines of Code (SLOC). Retrieved April 21, 2005 from the World Wide Web: http://www.dwheeler.com/.
30. Moglen E, Freeing the Mind : Free Software and the Death of Proprietary Culture. 2003.
31. Delio M, Linux: Fewer Bugs Than Rivals. Retrieved April 21, 2005 from the World Wide Web: http://www.wired.com/news/linux/0,1411,66022,00.html. 2004.
32. OpenSource, Peruvian Congressman refutes Microsoft's "Fear, Uncertainty and Doubt" (F.U.D.) concerning free and open source software. Retrieved April 21, 2005 from the World Wide Web: http://www.opensource.org/docs/peru_and_ms.php.
33. LinuxNews, Medicine's Dirty Software Secret. Retrieved April 21, 2005 from the World Wide Web: http://www.linuxmednews.com/linuxmednews/955063770/index_html. 2000.
34. Littlejohns P, Wyatt JC, Garvican L, Evaluating computerised health information systems: hard lessons still to be learnt. *BMJ*, 2003. 326(7394): p. 860-3.
35. Romanow R, Commission on the Future of Health Care in Canada. Building on Values: the Future of Health Care in Canada. 2002.
36. Lougheed T, Providing open access to research, one article at a time. Retrieved April 21, 2005 from the World Wide Web: http://www.cmaj.ca/cgi/content/full/172/5/621. *CMAJ*, 2005. 172(5).
37. Orbinski J, Access to medicines and global health: Will Canada lead or flounder? Retrieved April 21, 2005 from the World Wide Web: http://www.cmaj.ca/cgi/content/full/170/2/224. 2004.
38. Bloor R, South America warms to Open Source. Retrieved April 21, 2005 from the World Wide Web: http://www.theregister.co.uk/2005/02/10/south_america_open_source/ . 2005.
39. Reuters, MIT urges Brazil to adopt open-source. Retrieved April 21, 2005 from the World Wide Web: http://www.msnbc.msn.com/id/7220913/. 2005.
40. Wade T, MIT Backs Brazil's Free Software over Microsoft. Available from: http://www.eweek.com/article2/0,1759,1777134,00.asp. 2005.
41. TheEconomist, Friend or foe? Retrieved April 21, 2005]. URL:http://www.economist.com/business/displayStory.cfm?story_id=1699434. 2003.
42. BusinessWeek, Linux Inc. Retrieved April 21, 2005 from the World Wide Web: http://www.businessweek.com/magazine/content/05_05/b3918001_mz001.htm. 2005.
43. LinuxDevicesInc, 20 Linux Cell Phones in the Works. Retrieved April 21, 2005 from the World Wide Web: http://www.eweek.com/article2/0,1759,1765103,00.asp?kc=EWRSS03119TX1K0000594. 2005.
44. Prasad A, Oracle and Novell certify their enterprise Linux offerings. Retrieved April 21, 2005 from the World Wide Web: http://www.itp.net/news/details.php?id=13678&category. 2004.
45. Netcraft, Web Server Survey. Retrieved April 21, 2005 from the World Wide Web: http://news.netcraft.com/archives/2005/01/01/january_2005_web_server_survey.html. 2005.
46. Salkever A, Apple's Real Worry Isn't the Loss of IE. Retrieved April 10, 2005 from the World Wide Web: http://www.businessweek.com/technology/content/jun2003/tc20030618_7983_tc056.htm. 2003.
47. ComputerWeekly, Munich chooses Linux over Windows. Retrieved April 21, 2005 from the World Wide Web: http://www.computerweekly.com/Article122160.htm. 2003.
48. Broersma M, OpenOffice makes government inroads. Retrieved April 21, 2005 from the World Wide Web: http://news.com.com/2100-7344_3-5128730.html. 2003.
49. Becker D, Sun pushes OpenOffice standard. Retrieved April 21, 2005 from the World Wide Web: http://news.com.com/Sun+pushes+OpenOffice+standard/2100-1013_3-5390977.html. 2004.
50. Carrera D, The Future Is Open: What Open Document Is And Why You Should Care. Retrieved April 21, 2005 from the World Wide Web: http://www.groklaw.net/article.php?story=20050130002908154
51. Hermida A, Free software to aid poor doctors. Retrieved April 21, 2005 from the World Wide Web: http://news.bbc.co.uk/2/hi/technology/3331739.stm. 2003.
52. Oscar. Retrieved April 21, 2005 from the World Wide Web: http://oscarmcmaster.org/. 2004.
53. Raynux. Retrieved April 21, 2005 from the World Wide Web: http://www.rad.unipd.it/progetti/raynux/rayUK.php3.
54. Open Source Software for Better Software. Retrieved April 29, 2005 from the World Wide Web: http://www.euspirit.org.
55. GPCG, General Practice Computing Group Annual Forum Summary of Proceedings," GPCG (Australia). Retrieved April 21, 2005 from the World Wide Web: http://www.gpcg.org/publications/docs/Forum2003/ReportofProceedings.pdf. 2003.
56. Sullivan P, Purchase of software firm sign of medicine's electronic future: CMA. 2004.
57. Canadian Healthcare Technology. Retrieved April 29, 2005 from the World Wide Web:http://www.canhealth.com/jan03.html.
58. Copyright Law, US-GOV. Retrieved April 29, 2005 from the World Wide Web: http://www.copyright.gov/circs/circ1.html. 2000.
59. IDABC, France advances open source content management system. Retrieved April 29, 2005 from the World Wide Web: http://europa.eu.int/idabc/en/document/3131/335. 2004.
60. Copyright Law, US-GOV. Retrieved April 29, 2005 from the World Wide Web: http://www.copyright.gov/title17/.
61. Bonert M. Retrieved April 28, 2005 from the World Wide Web: http://individual.utoronto.ca/bonert/omr.html. 2005.
62. Kablet, France updates e-strategy. Retrieved April 21, 2005 from the World Wide Web: http://www.kablenet.com/kd.nsf/Frontpage/A50F0781A58E413F80256E370043B81F?OpenDocument. 2004.
63. Perera R, Open-source fans welcome French government move. Retrieved April 21, 2005 from the World Wide Web: http://archives.cnn.com/2001/TECH/industry/11/27/french.open.source.idg/. 2001.
64. Bundestag. Retrieved April 21, 2005 from the World Wide Web: http://www.bundestag.de/bp/2002/bp0202/0202100c.html. 2002.
65. WashingonPost, Microsoft Windows: Insecure by Design. August 23, 2003.
66. Naraine, R., US-CERT: Beware of IE. Retrieved April 21, 2005 from the World Wide Web: http://www.internetnews.com/security/article.php/3374931. 2004.