

# **CAN OPEN-SOURCE SOFTWARE REPLACE COMMERCIAL PRODUCTS IN MIS TEACHING? USING MYSQL IN AN INTRODUCTORY MIS CLASS**

**M. Nystedt, Abu Dhabi Women's College, Abu Dhabi, UAE**

## **ABSTRACT**

Open-source software is available for users to freely download, customize, and redistribute. To users, as well as developers, it offers benefits that traditional commercial software cannot easily match. Since the source-code is available, and it is not tied to, often restrictive, commercial software agreements, it is by some seen as more flexible and reliable. Open-source software, such as Linux, Mozilla, and OpenOffice.org, are becoming increasingly popular among commercial businesses, as well as academic institutions. This paper explores the benefits and drawbacks of the use of open-source software by academic institutions.

## **Introduction**

In commercial businesses, open-source software like Linux, Mozilla, and Openoffice.org is becoming increasingly popular. Educational establishments, like colleges and universities, have a long tradition of using a wide range of software, but in my experience, the use of open-source software has been limited. I decided to switch to MySQL from Microsoft Access for some of the MIS courses I teach, and this paper details that decision and my experience so far. This paper will specifically explore the use of the MySQL database server software for one introductory MIS class during spring 2004. It is not a technical paper, and it does not go into details about the MySQL software. Rather, I give an overview of open-source software, the MySQL technology, what I have used it for in my MIS class and my experiences so far.

## **Open-source software**

The term “open source” has its origin in 1998, when Eric Raymond after discussions with some of his colleagues, settled on the term “open-source software” instead of “free software” (Raymond, 1998). Open source is a category of software that seems to be growing in popularity, and it has established a foothold in many academic institutions (Kiernan, 1999; Pfaffenberger, 2000). The basic philosophy behind open-source software is that the source code is freely and openly available to anyone who would want to look at it, make modifications, and use it. Most commercial software is distributed in the form of compiled binary code with various types of licensing agreements. These licensing agreements typically restrict the use of the software in various ways, such as stating that you are not allowed to change it in any way, or that you are only allowed to install a set number of copies. Another popular form of software distribution, shareware, basically means you can try before you buy. In other words, users can acquire the binary copies of the software and try it out for a set period, or with limited functionality, and after that decide if they want to pay for it and keep it. Compared with these two models of software distribution, open-source distribution has the least restrictions. Users can typically download the source code from a web site, modify it in any way they want, create their own versions of the software, and even make that software available to other users. What you can and cannot do with open-source software depends on what license it is released under. Arguably, the most common license is the GNU General Public License (GPL, <http://www.gnu.org/copyleft/gpl.html>), often referred to as “copyleft”.

Open source could be considered a loose philosophy that has been present for over three decades. However, only recently has it been given a widely accepted name and reputation (Raymond, 1998, 2000). Allbritton (2003) called open source “a model and mechanism for collaborative knowledge construction”. Malloy, Jensen, Regan, & Reddick (2002) stated that open source “provides a model of collaborative knowledge construction that parallels that of science”. The Open Source Initiative (OSI), in their definition of open-source software, listed a set of rights that a software license must provide in order to be certified as open source (OSI, 2004):

“Free redistribution

Source code

Derived works

Integrity of author’s source code

No discrimination against persons or groups

No discrimination against fields of endeavor

Distribution of license

License must not be specific to a product

License must not restrict other software

License must be technology-neutral”

## **Open-source products**

Today, you can find open-source alternatives for pretty much any commercial software product you can think of, and for almost any device you can imagine. For example, OpenOffice.org (<http://www.openoffice.org>) is one of the best-known open-source software products. It is an

office productivity suite, which in many ways can rival the clear market leader Microsoft Office. Most of the functions you will find in Microsoft Office are also available in OpenOffice.org and, for most users OpenOffice.org would be a perfectly acceptable alternative, including users at colleges and universities. Although OpenOffice.org is a stable, mature, software product with all the functionality the average user would need, its biggest selling point is that it is available free. When Forrester Research looked at 140 “major North American companies” they found that OpenOffice.org is currently used in 12 of the companies, and eight more are planning to use it in the future (Krim, 2004). When asked about open-source software in total, 65 of the companies said they are using open-source software and an additional 20 said they are planning to in the future.

Other prominent open-source software products include Linux, Mozilla, Apache, and the GIMP. Arguably, the best-known open-source software, Linux, started out in 1991 as a personal quest for Linus Torvalds in Finland, to write his own operating system. It has grown into a commercially competitive platform which is today in use in many businesses and organizations and available for PDAs, cell phones, and set top boxes, etc. In a study of several hundred businesses of less than 1,000 employees, Jupiter Research found that 19% were using some form of Linux operating system on their desktop computers (Cowley, 2003). Cowley attributed this popularity primarily to the open-source software being freely available. The Economist (1998) noted that Linux was already in 1998 becoming a serious rival to the dominant Microsoft Windows operating system. More recently, The Yankee Group found that around a third of U.S. businesses plan to migrate at least some Microsoft Windows computers to Linux (Becker, 2004). When Netscape released the source-code to their web browser suite Communicator as open source in 1998, one result was the collaborative open-source effort Mozilla (<http://www.mozilla.org>). Mozilla has evolved into a mature and very popular web browser suite, including a web editor and email client. Apache is the most popular web server software in use on the Internet today. According to <http://www.netcraft.com> on 8/30/2004, 67.7% of the surveyed 53 million publicly available web sites were running Apache, with Microsoft’s Internet Information Server (IIS) a distant second at 21.2%. In fact, the acronym LAMP (Linux, Apache, MySQL, PHP/Perl/Python) denotes a group of open-source software that has had a significant effect on the development of dynamic web applications. The GIMP (<http://www.gimp.org>) is a high-quality graphics editing software that in many ways can rival the dominating commercial software Adobe Photoshop. Version 2.0 of the GIMP was released in March 2004 for UNIX, Windows and Mac OS X simultaneously.

There is no doubt that open-source software is a major movement today. At <http://www.sourceforge.net>, one of the primary communities for open-source software development, around 80,000 different projects are listed. The vast majority of those projects are only of interest for a very small community, but many of them are viable alternatives to commercially available software. Corporate America is paying serious attention to open-source software. IBM, for example, has for many years, been a strong proponent of Linux and is now considered by many to be the largest distributor of Linux in the world. In fact, in August 2004, IBM announced that they would donate software valued at \$85 million to the Apache open-source

group (Lohr, 2004). Even the U.S. Federal government is becoming a major user of open-source software. For example, the National Security Agency (NSA) has released a version of Linux they call “Security-Enhanced (SE) Linux” (see <http://www.nsa.gov/selinux/>). In addition, the State of California has, in the latest budget-crunch turned to open-source software as a way of cutting costs (CPR, 2004).

### **Benefits and drawbacks of open-source software**

Being available free may be the most obvious benefit of open-source software, but besides being free, open-source software has a number of advantages compared to commercial software:

- Since the source-code is available for users to download and modify, users can create their own customized applications to suit their unique needs.
- Although this is a highly debated point, many view open-source software as more secure than commercial software because the source code is available openly to anyone. Software projects like Mozilla have literally thousands of users contributing to the development and thus problems with the software can be fixed much faster (Newmarch, 2001).
- Open-source software is very often available in versions for different platforms. Mozilla, for example, is available for Windows, Mac, and Linux, as is OpenOffice.org. Many open-source projects originate in the Linux community, which makes them relatively easy to port to other operating systems like, for example, Mac OS X, which at the core is FreeBSD 5 (see <http://www.apple.com/macosx/features/unix/>).
  - Open-source software is free from lock-in. If you buy a piece of commercial software, some argue, you are likely to be tied in to that technology with no possibilities to switch to other software. You are also vulnerable to companies discontinuing software or closing down business completely. The argument for open-source software is that since the code is openly available, the software can live on forever. Jesiek (2003) argued there is little question that open-source communities have created many software applications that are more flexible and adaptable when compared to similar commercial offerings.
  - Open-source software often has lower hardware requirements than commercial software. You could use MySQL and Linux on hardware that would not be powerful enough to run commercial alternatives like Microsoft Windows 2003 Server and Microsoft SQL Server. An old discarded Pentium II or III computer, for example, could with enough RAM and hard drive space, function perfectly well as a web- and database server for a large number of students.
  - In an educational context, Allbritton (2003) pointed out that the use of open-source software allows for greater freedom, flexibility and control for instructors. There is no need to wait for information technology support to install the software and you can test-drive it before fully implementing. This can also result in increased freedom and accessibility for the students. All the software students need for a course can be downloaded and installed on their own computers. Since the source-code is available to

the students, it also offers the opportunity for students to take a look “behind-the-scenes” at what is really happening.

The main criticism of open-source software is ‘you get what you pay for’. For example, Microsoft often makes the case, that when you buy one of their products you get support and other assistance with the purchase. While it is true that you do not typically get manuals with open-source software, or a phone support number you can call if you have problems, there are extensive communities online where you can get help. In addition, for the more popular open-source software products, like Mozilla and OpenOffice.org, there is a multitude of literature that you can purchase and use for training and support. Also, some open-source communities are experimenting with various ways of offering support. Mozilla, for example, has a telephone support number, which you can call, and for \$39.95 per case, you can get professional tech support in much the same way you would by calling any commercial software developer. It should perhaps be noted, that it seems to be an increasing trend even in commercial software to not provide printed documentation with the product. Instead, you receive it in electronic format and can print it if you want to. It is important to point out, that you should not expect that “free” equals “no cost”. You may not have to pay a price for purchasing the open-source software, but you will incur costs for support, development, etc. Some would argue that those costs are greater than what you would pay for the commercial software license up front.

Other common criticisms of open-source software include that it is not user-friendly and that it is technology-focused (Levesque, 2004). It is true that if you browse around the selection of open-source software at SourceForge.net, you will find many applications that are far from polished when it comes to the user interface and the user experience. Nevertheless, it is also true that major open-source offerings, like Mozilla and OpenOffice.org, have sophisticated and user-friendly interfaces that can rival anything commercially available. That said, many open-source projects do tend to focus on the latest and coolest technology, instead of bringing end-users increased value. In essence, programming for themselves instead of users, is commonplace in the open-source community. However, that is how many innovations in software design come about.

Finally, it is also worth noting that the open-source philosophy has spread to areas other than software, including open courseware at MIT (<http://ocw.mit.edu>) and the Creative Commons license (<http://creativecommons.org>), applicable to various kinds of intellectual work. These efforts usually fall under the umbrella-term “open content” which Cedergren (2003) defined as “content provided not-for-profit – often collectively – with the intentional purpose of making content available for further distribution and improvement by others at no cost”.

## **MySQL**

One of the best known and most widely spread open-source software applications is MySQL. MySQL is an open-source database server product that is being developed and marketed by

MySQL AB in Sweden. The MySQL software has its origin in a database system called UNIREG, created in 1979 by one of the founders of the company. MySQL AB, the company, was founded in 1995 and their main product, the MySQL database server, was released in 1996. It has since then steadily evolved into a mature software product that is now in use in many commercial organizations. Major businesses, for example Yahoo!, Google, Cisco, NASA, and Dunn & Bradstreet, rely on MySQL software for their daily activities (MySQL, 2004). According to MySQL AB, there are today over four million active MySQL installations worldwide and the server software is downloaded over 35,000 times per day (MySQL, 2004). Other major users of MySQL, according to Kirkpatrick (2004), include Sabre, the travel reservation system. Kirkpatrick (2004) also reported that SAP in 2003 began to recommend MySQL to its customers and other giants, like Veritas and BEA, have aligned themselves with MySQL. Another development that is sure to increase MySQL's exposure to corporate customers is the availability of cluster technology (Krill, 2004). Hall (2003) reported that MySQL is also starting to make inroads in corporate data centers, where performance, in addition to price, is often the major selling point.

Although MySQL is available free for anyone to download and use, many businesses pay the \$495 per server per year fee to the Swedish company. Why would they do that? According to Kirkpatrick (2004), Amazon, Cisco Systems, and Sabre, for example, do that for two reasons: they want the company - MySQL AB - to stand behind their product and offer service, and they get the right to make changes to the MySQL code and incorporate that into their own products. MySQL is distributed under the GPL, which states that you can freely download and alter software but if you re-distribute it, your software also has to be distributed under the GPL, something that does not always suit commercial businesses.

Compared to some of its commercial competitors, MySQL runs very well on relatively simple hardware. The MySQL server I have used for the activities discussed in this paper, is running on a Dell PowerEdge server with an Intel Xeon 1.8Ghz processor and 1Gb RAM. At the most, we have had twenty concurrent users, connected from on and off campus, and from the users' perspective there was no noticeable slowdown. The Dell server is running Microsoft Windows 2003 Server software, and in addition to being a MySQL server, it works as a FTP-, web- and VPN server. Since the server was already in use before we started using MySQL, we just went with the Microsoft operating system. If we were starting from scratch, we would definitely consider going with open-source software instead.

The MySQL server's native interface is only command-line based, like so much other software with roots in the UNIX world, so for beginners it is not user-friendly. However, there are a number of graphical-user interface (GUI) clients available that can be used as front-ends to the server, including Microsoft Access. I decided to recommend to the students the MySQL Control Center, which is a free product from MySQL AB (see <http://www.mysql.com/products/mysqlcc/>). The Control Center gives a user-friendly, GUI interface, to working with databases on a server. It involves an environment the students are familiar with, i.e. their operating system, which means it is more usable for them. They can within a relatively familiar environment, for example, create

databases and tables, enter data, and create and run queries. The MySQL Control Center, although still in beta (0.94 as the time of writing this), is available for both Windows and Linux. My favorite client for Mac OS X is CocoaMySQL (<http://cocoamysql.sourceforge.net>), also an open-source project. There is also a GUI interface for administering a MySQL server, the MySQL Administrator. Both the Control Center and the Administrator are open source, released under the GPL, and available for both Windows and Linux with Mac OS X versions coming in the future.

MySQL has been used in academia for some time. For example, Triveid, Kar, & Patterson-Neill (2003) developed an assignment management and peer evaluation system using MySQL as the backend database manager. They did not discuss the reasons for choosing MySQL in particular, but stated “the system has been developed by making use of free distribution of software products available for the public”. Meyer & Conry (2002) developed a new course in building web applications, using MySQL as the database manager on the web server. They also created a CD-ROM with MySQL and other free software on it, including PHP. That would have been very hard to do with commercial software. Chung & McLane (2003), for economic reasons, decided to pick MySQL for a client/server-programming course they developed. Allbritton (2003) used open-source software, including MySQL, PHP, and Perl, to teach introductory computer programming skills and web development to psychology students.

Through all my teaching at college and university, the database of choice has always been Microsoft Access. At one university I worked for, Oracle’s database product was used for a higher-level database and systems administration course, but that is the one exception in my career so far. I wanted to explore the use of some alternative for my courses, for primarily three reasons:

Microsoft Access is commercial software. The MySQL server is free, and there are many free client applications that you can use. Budgets are tight in higher education and students are well known for not having much money. Lower cost should be of interest to both parties.

Microsoft Access is not client-server software. You can use Microsoft Access as a limited database server, but you do not even come close to some of the features on more advanced database systems, such as Oracle or IBM DB2. MySQL may not have all the features of those systems, but it is much closer to a mature, client-server database system than Access is.

Microsoft Access is only available for Microsoft Windows. The MySQL server can run on a wide range of platforms, including Linux, Microsoft Windows, and Mac OS X. To access and work with a MySQL server you can use different kinds of software available for pretty much any operating system in existence today. This gives flexibility and choice to both teachers and students.

### **Using MySQL in my MIS class**

During spring semester 2004, I used MySQL in two of my classes: MIS327 Information Systems Fundamentals, and MIS337 Business Systems Analysis and Design. My first use of the MySQL system discussed in this paper was with MIS327 during fall 2003.

MIS327 introduces students to the field of MIS. One assignment in the course is for students to, given a simple scenario (car rental business, library, or similar), create an entity-relationship diagram of that scenario (typically around five entities), and implement that diagram as a database on the MySQL server. Since this is the introductory class, we are not too strict on the correctness of the entity-relationship diagrams, but we do require the students to transfer the diagrams correctly to the database. In MIS337, we are stricter in terms of judging the models that the students create, and the databases are more complex compared to MIS327. Figure 1 shows an actual entity-relationship diagram from the MIS327 class.

### **Figure 1 - Example entity-relationship diagram**

Installation and maintenance of the MySQL server was surprisingly straightforward and pleasant. I downloaded the software from the MySQL web site, and installed it with only minor problems. Configuring the server software and user accounts was also very straightforward and went without major issues. I created a unique user account on the MySQL server for each student, and I also created a database for each student, where they would create the tables for the assignments. All students also have access to a test database where they can practice MySQL without harming the database they submit for the course.

Using the MySQL server's native command-line interface, you would create a database using the following command:

```
mysql> CREATE database dvdrental;  
Query OK, 1 row affected (0.00 sec)
```

And to create a table, you would enter the following:

```
mysql> USE dvdrental;  
mysql> CREATE TABLE customer (  
-> customerID int not null,  
-> customerName varchar(40) not null,  
-> customerPhone char(10) not null,  
-> customerAddress varchar(30) not null,  
-> customerEmail varchar(30) null,  
-> );
```

Obviously, this is not very user-friendly, especially for users who are not very experienced with computers and databases, so I decided to recommend the use of the MySQL Control Center. You



can see an example of the Control Center interface in Figure 2.

## **Figure 2 - MySQL Control Center Screen shot**

When I first used this approach in one of my classes, a concern I had was how students would take to it, and if it would be hard for them to understand the software. As it turned out, they have found the Control Center relatively easy to understand and work with. In fact, in the database assignments I have issued, the deductions most students get are for mistakes with the entity-relationship models, and not with mistakes in the actual databases. To help students get familiar with the Control Center, I wrote a brief tutorial (see <http://www.nystedt.org/misc/mysqlcc/mysqlcc.html>) that covers the main skills they need for the assignments. It is a self-study tutorial, which students can work through on their own.

The students' response to using MySQL has been overwhelmingly positive. Students especially liked that they could access their database from wherever they are. In the group projects for MIS337, this was especially true, because they could form a central repository on the server and everyone could access it. Since many students were familiar with Microsoft Access, they said they were at first a bit frustrated with learning a new piece of software, but after they had worked with it for a while, the general view was that they appreciated learning something new, and the MySQL Control Center was easy to work with. One student told me that he wish that we would use much more of this kind of software in our courses, since it had given him a lot broader perspective on systems and software. The same student also said he had mentioned MySQL in a recent job interview and that the interviewers told him that they thought that was a good experience to have.

This approach has worked out very well for us, and here are some of the main benefits:

- Students work on their databases on the server, and when the submission deadline passes, I simply cut off access to the server, so they cannot go in and change anything after that. In the past, when I have issued similar assignments, I have usually asked students to hand in the database file(s) on disk, which is both cumbersome for me and cost money for them. All of that is eliminated by using MySQL.
  - As discussed earlier, students can work from anywhere as long as they have a computer and internet access. They can do the work on campus, our School of Business computer lab, their dorm room, from home, or their place of work. This allows them flexibility that using Microsoft Access alone cannot offer.
  - Students will get exposure to MySQL in our introductory MIS class and we can take advantage of this in other classes. For example in MIS337 where we demand more of students in terms of models and database complexity, and in other courses on web programming, networking and databases.
  - Using MySQL for these courses have widened the students' experience and that

is, in our view, always a good thing.

- The fact that the software is available at no cost is obviously a significant benefit. So far we have not experienced any problems with the software that could not be solved with the help of books and online resources.

Some of the drawbacks of the approach presented in this paper:

- MySQL is not as easy to use as Microsoft Access (this is especially true for creating relationships which is just drag and drop in Access), even when using a GUI client. This has more to do with the client software than the MySQL server in it self. As the client software develops, I am sure user-friendliness is a primary concern. It should be pointed out that it is not necessarily a choice between MySQL and something like Microsoft Access. Although there is the MySQL Control Center, MySQL is primarily the database server and Microsoft Access could very well be used as the client software used to connect to the server (e.g. Kaufman, 2002).
- I brought this up as a benefit, but it can also be a drawback. Users need to be online, i.e. connected to the Internet, to work with the database server. Although there are clear advantages in giving students flexibility in working wherever they want to, it also requires students to have Internet access and that the server is up and running. This is a diminishing problem in our view, since more and more students do have Internet access from their homes and places of work.
- Support can be an issue since just downloading MySQL does not give you any. MySQL AB offers training courses you can pay for, and they also sell support packages in various levels for \$1,500-\$48,000 per year depending on how much and what kind of support you require. There is a growing body of books, you can purchase to help you with many aspects of setting up, running, and using a MySQL system. One book I can recommend is Vaswani (2004).
- Security is of course always a concern, and by installing MySQL, you have to have a server and open it up to the rest of the world. You can of course deny access to it for anyone outside your campus, but that defeats the purpose of allowing users to connect from anywhere. Our experience has so far been that as long as you manage the security of the server computer in an appropriate way, there should not be any particular security problems with running a MySQL server. However, it is a good idea not to run any critical systems on the same machine as a MySQL server you use for teaching. You should also consider putting the server on a separate network segment, if possible, to better control network access and traffic.
- What may be a downside for some faculty is that at most institutions when choosing to go for an open-source alternative, you are also setting yourself up to become your own systems administrator, helpdesk, and all-around support function. Most colleges and universities will not support systems and software that is not on the official list, and that fact alone may turn some away because of added workload. This could of course also be viewed as a benefit because if you are interested in the technology and willing to put in the time, you are in for an exciting learning experience.
- Another potential downside is that students often want to learn software they

know of, which in the area of databases is typically Microsoft Access. My view is that it is more important for students to learn the fundamentals than it is for them to learn a specific software application.

## Conclusions

The title of the paper is perhaps a bit misleading since the question is not really if open-source software can replace commercial software in teaching and research, but if it can compliment it. I think the answer to that question is a resounding yes, and I would encourage more educational institutions to look at open-source software like Linux, MySQL and OpenOffice.org. For us at the School of Business at Francis Marion University, the trial with MySQL has so far turned out to be positive and I foresee us continue expanding our use of MySQL, and also look at other open-source software for use in our MIS teaching and research. This does not mean, however, that we will completely abandon using commercial software.

## Acknowledgement

I would like to acknowledge students in my MIS327 and MIS337 classes, Spring 2004, who gave me feedback on the use of MySQL and helped me find some of the references used in this paper.

## References

- Allbritton, D. W. (2003). Using open-source solutions to teach computing skills for student research. *Behavior research methods, instruments, and computers*, 35(2), 251-254.
- Becker, D. (2004). *Switching to Linux picks up steam*. Retrieved 8/30, 2004, from <http://news.com.com/2100-7344-5330340.html>
- Bezroukov, N. (1999). *Open source software as a special type of academic research (a critique of Vulgar Raymondism)*. Retrieved 03/24, 2004, from [http://firstmonday.org/issues/issue4\\_10/bezroukov/](http://firstmonday.org/issues/issue4_10/bezroukov/)
- Bray, H. (2003, Dec 22). Open-source battle is heating up. *The Boston Globe*, p. C3.
- CATAAlliance. (2004). *Industry Pulse Survey Shows Open Source is Becoming Part of Mainstream IT Strategy*. Retrieved 05/01, 2004, from [http://www.cata.ca/Media\\_and\\_Events/Press\\_Releases/cata\\_pr04210402.html](http://www.cata.ca/Media_and_Events/Press_Releases/cata_pr04210402.html)
- Cedergren, M. (2003). Open content and value creation. *First Monday*.
- Chung, W. S., & McLane, D. (2002). Developing and enhancing a client/server programming for Internet applications course. *The journal of computing in small colleges*, 18(2), 79-91.
- Claybrook, B. (2004). *The business value of open source*. Retrieved 08/10, 2004, from <http://www.linuxworld.com/story/45707.htm?DE=1>
- Cowley, S. (2003). *Small businesses like Linux prices: Jupiter study finds growing interest in open-source*

*alternatives*. Retrieved 03/24, 2003, from <http://pcworld.shopping.yahoo.com/yahoo/article/0,aid,111616,00.asp>

CPR. (2004). *Explore Open Source Alternatives*. Retrieved 08/12, 2004, from <http://www.report.cpr.ca.gov/cprprt/issrec/stops/it/so10.htm>

Economist. (1998). *Hacker journalism*. Retrieved 04/14, 2004, from [http://www.economist.com/displaystory.cfm?story\\_id=265022](http://www.economist.com/displaystory.cfm?story_id=265022)

Feller, J., & Fitzgerald, B. (2001). *Understanding open source software development*. London: Addison-Wesley.

Hall, M. (2003, Oct 13). MySQL breaks into the data center. *ComputerWorld*, p. 32.

Jesiek, B. K. (2003). Democratizing software: open sources, the hacker ethic, and beyond. *First Monday*.

Kaufman, S. (2002). *How to use Access 2000 as a database front end*. Retrieved 03/15, 2004, from <http://builder.com.com/5100-6388-1050135.html>

Kiernan, V. (1999). *The 'open-source movement' turns its eye to science: revealing the code for software used in research will help verify data, advocates say*. Retrieved 05/01, 2004, from <http://chronicle.com/free/v46/i11/11a05101.htm>

Kirkpatrick, D. (2004, Feb 23). How the open-source world plans to smack down Microsoft, and Oracle, and ... *Fortune*, 149, 92.

Krill, P. (2004, Mar 12). MySQL cluster gets ready to ship. *InfoWorld*.

Krim, J. (2004, March 19). Microsoft is facing long-term adjustment: firm's basic strategy could be disrupted. *Washington Post*, p. E01.

Levesque, M. (2004). Fundamental issues with open source software development. *First Monday*, 9(4).

Lohr, S. (2004, August 3). In competitive move, IBM puts code in public domain. *New York Times*.

Malloy, T. E., Jensen, G. C., Regan, A., & Reddick, M. (2002). Open courseware and shared knowledge in higher education. *Behavior research methods, instruments, and computers*, 34, 200-203.

MySQL. (2004). *MySQL company fact sheet*. Retrieved 3/23, 2004, from <http://www.mysql.com/company/factsheet.html>

Newmarch, J. (2001). Lessons from open source: intellectual property and courseware. *First Monday*.

Nichols, D. M., & Twidale, M. B. (2003). The usability of open source software. *First Monday*, 8(1).

OSI. (2004). *The open source definition*. Retrieved 08/30, 2004, from [http://www.opensource.org/docs/definition\\_plain.php](http://www.opensource.org/docs/definition_plain.php)

Perens, B. (1999). The open source definition. In C. DiBona, S. Ockman & M. Stone (Eds.), *Open sources: voices from the open source revolution*. Sebastopol, CA: O'Reilly & Associates.

Pfaffenberger, B. (2000). *Linux in higher education: open source, open minds, social justice*. Retrieved 05/01, 2004, from <http://www.linuxjournal.com/article.php?sid=5071>

Raymond, E. (1998). *Goodbye, 'free software'; hello 'open source'*. Retrieved 08/12, 2004, from <http://www.catb.org/~esr/open-source.html>

Raymond, E. (2000). *The cathedral and the bazaar*. Sebastopol, CA: O'Reilly & Associates.

Schweik, C. M., & Semenov, A. (2003). The institutional design of open source programming: implications for addressing complex public policy and management problems. *First Monday*.

Stalder, F., & Hirsh, J. (2002). Open source intelligence. *First Monday*, 7(6).

Stallman, R. (1992). *Why software should be free*. Retrieved 03/23, 2004, from <http://www.gnu.org/philosophy/shouldbefree.html>

Syllabus. (2004). *A balancing act? Openness and security on campus*. Retrieved 04/06, 2004, from

<http://www.syllabus.com/article.asp?id=9193>

Trivedi, A., Kar, D. C., & Patterson-McNeill, H. (2003). Automatic assignment management and peer evaluation. *The journal of computing in small colleges*, 18(4), 30-37.

Vaswani, V. (2004). *MySQL: The complete reference*: McGraw-Hill/Osborne.