2003

# An Integrated, Secured, Open-Source Medical Prototype for Collaborative Patient Management on the Internet

Joseph Dabre Ramsey
*Nova Southeastern University*, ramsey2785@gmail.com

# An Integrated, Secured, Open-Source Medical Prototype for Collaborative Patient Management on the Internet

by

Joseph Dabre Ramsey

A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy

Graduate School of Computer and Information Sciences
Nova Southeastern University

2003

We hereby certify that this dissertation, submitted by Joseph Dabre Ramsey, conforms to acceptable standards and is fully adequate in scope and quality to fulfill the dissertation requirements for the degree of Doctor of Philosophy.

[illegible] 7/30/03

John Scigliano, Ed.D. Date
Chairperson of Dissertation Committee

[illegible] 7/23/03

Jacques Levin, Ph.D. Date
Dissertation Committee Member

[illegible] 7/21/03

Junping Sun, Ph.D. Date
Dissertation Committee Member

Approved:

[illegible] 7-30-03

Edward Lieblein, Ph.D. Date
Dean, Graduate School of Computer and Information Sciences

Graduate School of Computer and Information Sciences
Nova Southeastern University

2003

An Abstract of a Dissertation Submitted to Nova Southeastern University in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy.

An Integrated, Secured, Open-Source Medical Prototype for Collaborative Patient Management on the Internet

by
Joseph Dabre Ramsey

June 2003

Conventional approaches to building critical and secured systems are based on the use of commercial tools for development and maintenance. Changes in the marketplace and the acceptance of the open-source model have brought this assumption into question. The combination of open-source's rapid rise and the introduction of pervasive computing has made the computing industry more receptive to open-source tools and products. The open-source model allows systems to be controlled by a single individual or a small developer group that reduces dependence on individual experts. The availability of free system source codes, an expanding commercial support market, and increasing global collaborative projects makes open-source an important development in the computing environment and an exciting innovation in software engineering.

Open-source projects require a level of modeling to successfully implement a solution. This study implemented a Web application prototype that models medical business logic and state that is secured. The researcher adopted the object-oriented design methodology and prototyping that improved security and lowered overall development cost. The open-source community had played an increasingly significant role in the business plans of established computing companies, in university research labs, and in the development of new companies focused on open-source support and integration issues.

The openness of the Internet presents both system development and privacy issues. The availability of free tools and instructions on how to compromise systems is alarming within the online community. Thus, open-source security tools are helping protect people's privacy by enforcing authentication, confidentiality, and information integrity to prevent unauthorized access. Open-source growth motivated this research to develop a medical prototype for online collaboration. Open-source tools including PHP, MySQL, Apache Web Server, and the Linux operating system were used to develop the secured application through prototyping.

The main contribution of this study is that it demonstrated the exclusive use of open-source software and tools for an online application. The researcher hypothesized that open-source tools like PHP, MySQL, XML, and LINUX are the answer to building dynamic multi-tiers and cost effective systems faster. The research also explored major tools available for open-source software development.

## Acknowledgments

A dissertation requires extensive support from your faculty, family, friends and the general social community. There are many people deserving acknowledgement for their roles in helping me through this work. Foremost, I thank my advisor, Dr. John Scigliano for his tireless and persistent patience. Working with Dr. J. Scigliano, Dr. J. Levin, and Dr. J. Sun, I became precise, and able to carry logical arguments to conclusions. Many thanks to my family, especially my wife, Jean V. Ramsey, my daughter Martenni Ramsey, and son Kanzoni Ramsey for the support and patience they endured throughout the dissertation process. Thanks to the extended family in Detroit, Dr. Asabigi, Dr. Calvin Trent, Kim Trent, Jeffrey Trent, Raina Trent, Debbie Trent, Ken Coleman and to all the Trent's family in Detroit, Michigan for all the advice and help over the years. Many thanks to my family in London, England for the help in proof reading my dissertation and their encouragement, notable Issabelle Acharana, Veronica Acharana, Amina Asabigi, Ben and Anne Acharana. As an Adjunct Professor and the Information Technology Manager for the Detroit Water & Sewerage Department (DWSD), many people have influenced and helped in various settled ways. I will like to thank again my advisors at Nova Southeastern University and the faculty of Wayne State University's Computer Science Department Dr. Farshad Fotohi, Dr. Grosky –now at the University of Michigan, Dr. Stomp, Charles Alexander, and Alex George for the encouragement and mentoring on how to be a good professor. Many thanks to Sean Werdlow - Chief Financial Officer, City of Detroit; Ali Freeman - Deputy Director, Human Resources; Mart Grady, Deputy Director, Finance; Dave Rayford, Chief Information Officer; Lance Mitchell -DWSD; Betty Afari –DWSD, and Ken Yopp for their show of friendship throughout the period. My final thanks goes to my family in Ghana, Africa including Issa Sadiq, Regina Sadiq, Kanzoni Asabigi, John and San Asabigi and the many Kanzoni's, Asabigi's and all the extended family in Tamale, Walewale, Arigu, and Accra in Ghana. My final acknowledgement goes to the late Dr. Charles H. Wright for all the advice and guidance.

## Table of Contents

## List of Tables

### Tables

## List of Figures

**Figures (continued)**

# Chapter 1

## Introduction

The Internet has brought the online community both information as well as convenience. The Internet is the largest network of networks that has brought people together globally by allowing the proper set of network requirements for online collaboration (Gordon & Loeb, 2001). It uses Transmission Control Protocol/Internet Protocol (TCP/IP) and packet switching to facilitate an extensive communication environment. At the same time, the privacy of people and organizations is threatened on the Internet. Security has been a problem for online users. The development of perspectives and techniques necessary for a risky and malicious computing environment like the Internet calls for the use of cost-effective tools and design techniques. Thus, Internet security is necessary to protect privacy. Crucial information transmitted through the Internet needs to be protected from malicious attacks.

The Internet afforded unlimited opportunities for people to cooperate on projects without the limits of geographical borders. Online collaboration has allowed computer scientists to leverage technical resources to produce software applications such as the Linux operating system. Linux, an open-source operating system, was started in 1991 by Linus Torvalds, and continues to expand today through a community of global computer science volunteers leveraging the Internet (Drummond, 2000). Software applications produced by the online development community have led to the open-source paradigm.

The open-source paradigm remains a powerful collaborative technique of shared knowledge and research to meet common computing needs (Raymond, 2001). The paradigm advocates a community concept that provides online collaboration and creative diversity without geographical borders (Curtis, 1992; Retting, 1990). The primary focus for the open-source community had been to develop a wide range of free information management applications and tools. Open-source applications have evolved through the years and include operating systems, databases, scripting languages, and security tools that are operational in many computing environments.

The use of open-source tools in developing secured systems provides a multitude of benefits. These include a common set of tools that benefits consumers in cost, security, and stability (Garber, 2000). The investigator implemented an online medical system design using open-source tools. The prototype used the object-oriented methodology, a systematic software development approach that uses objects and notations to express underlying business concepts (Blaha, 2002). The investigator also used prototyping to build the online medical application that allowed the end-user's participation and evaluation throughout the development process (Laudon & Laudon, 1996).

## Problem Statement

The problem addressed in this study involved the security of sensitive medical data. The field of security and software development strategies is changing. The questions of technical quality, commercial package costs such as interfaces and training, increased vulnerability from network attacks, and internal cost pressures have forced companies to look at alternatives such as open-source. Open-source tools are superior and robust (Neumann, 1999). Closed-source proprietary software has been considered the lifeblood of computer systems; however, there are associated risks with proprietary software. According to Open Source Initiative (2001, p.2), Gerald P. Weinberg once famously observed that, "If builders built houses the way programmers built programs, the first woodpecker to come along would destroy civilization." The reliability of software has been poor. The most often-mentioned risks include unavailability of source code and limited peer analysis to improve reliability (Neumann, 1999).

The scale and ease of compromising sensitive data have caused concerns regarding access rights in open environments like the Internet. Confidential records and reports of terminal illness, domestic violence, psychiatric treatments, and alcoholism can be compromised if access controls are not adequately implemented for medical data accessed over the Internet. Details of important data about public and private lives of individuals are made available for online consumption. The range of privacy violations is staggering; the results and ramifications are unpredictable (Meeks, 1997). The security solutions on privacy addressed in this study concern the use of open-source tools to protect privacy and control access to medical resources over the Internet as a collaborative environment.

One of the troublesome aspects of the implementation of medical data in a secured collaborative environment is that most providers and carriers are distributed geographically. In a study by the National Institute of Standards and Technology, investigators Poole, Barkley, Brady, Cincotta, & Salamon (1999) indicated that a roadblock to efficient health care was the distribution of important information across multiple sites. Poole et al. concluded that these distributed sites, located across significant areas, caused the problem of the lack of a uniform mechanism to integrate medical information.

Another research problem addressed by this study is access security. The information on the Internet remains generally volatile, easy to manipulate, and at the same time accessible to many (Moor, 1997). Denial of service attacks such as the *Love*

*Bug* virus in the spring of 2001 showed an example of an insecure and vulnerable system. Building reliable, secure systems on the Internet remains elusive (Weaver, Vetter, Whinston & Swigger, 2000). Weaver et al. found that progress with Internet security was improving, but total prevention remains far away. The Internet had increased security vulnerability through the availability of penetrating tools, information on their use, and the lack of a sense of responsibility among the computer-literate who view any system as fair game (Lukasik, Greenberg, & Goldman, 1998).

Ahn, Sandhu, Kang, & Park (2000) observed that most existing Web-based workflow systems provide minimal security services such as authentication of users. Security remains the most important concern in the growing open-source community. Undetected attacks can strike web-based applications at any time, from anywhere, and in any form. New security issues arise every day, thereby threatening Web systems all over the world. As Internet usage proliferates, security becomes both important and complex. Akker, Snell, & Clement (2001) determined that contemporary users and systems were ill-equipped to deal with the complex security demands of an ubiquitous and insecure network like the Internet. There are prevailing temptations to use medical information for business and other purposes. For example, the abuse of genetic information is not far-fetched if accessible over the Internet. A recent study reported 206 cases of direct discrimination in employment and insurance problems from unauthorized use of genetic test information (Geller, Alper, Billings, Barash, Beckwith, & Natawicz, 1996).

Security remains a fundamental component of the Internet, and the assumptions of good security partly explain the rapid growth and adoption of intranets and/or extranets around the world (Goncalves & Brown, 2000). Because there are many paths and ways to get around the Internet, it is difficult to monitor or control Internet traffic on a site, opening the site to attacks from thousands of potential users. Directed denial of service attacks are typically the sort of intrusions targeted at single Web sites. Such assaults had shut down business on Amazon.com and eBay for hours, costing an estimated $1.2 billion in market capitalization, according to the Yankee Group. New research by the Yankee Group indicated that nearly 10 million U.S. homes would be digitally remodeled with home networking in the next four years (Spiegel, 1999), raising further the Internet's security spectrum.

The traditional software development process or waterfall approach is not suitable for online collaboration because each phase is normally completed before the next phase is started. Blaha (2001) described the waterfall approach as inappropriate for applications with substantial uncertainty in the requirements. The problem is that, for the waterfall model to be effective, a clear statement of requirements is necessary before design and implementation can start (Boehm, 1981). The waterfall approach, according to Verner & Cerpa (1999), fails in comparison with prototyping. Boehm (1988) pointed out that the waterfall approach was appropriate when the system is well understood and the risks of changing requirements were low. Online systems are historically security sensitive because of the dynamic nature of the Internet. Most online systems' requirements are dynamic and may require rapid development such as prototyping.

The risk of changing requirements may not be the only reason to avoid the waterfall approach. Another reason is that most modern commercial online applications are not built so much as they are evolved internally, with multiple iterations of the system being produced as features are added (McCarthy, 1995; McConnell, 1996). Prototyping allows flexibility of design, functionality of system components, and detection of problems. The traditional development process is very cumbersome because of the stages the developer must follow, including formal sign-offs and development team differences.

The waterfall model fails as a valuable development tool for online collaboration where the requirements are uncertain and a significant amount of risk is associated with implementation (Boehm, Gray, & Seewaldt, 1984). Also, commercial software packages are not suitable for online development because most packages are pre-coded. Software packages are geared to common requirements and are cumbersome to customize to meet specific goals. Commercial software packages typically do not provide formal specifications along with their software applications. With the lack of specifications, it would be difficult for developers to write specifications for commercial applications when they do not have access to the internals of the products. It is also likely that their efforts may be invalidated by changes and new versions at the manufacturer's will.

## Goal

The goal of the investigator in this study was to select a development strategy, design a secure system for medical collaboration on the Internet with multiple access security, and identify appropriate tools. The investigator included an overview of an open-source development environment and examined how open-source tools can be structured to the needs of three-tier development over the Internet. The investigator reviewed the features of open-source tools such as Linux, (operating system), MySQL, (object-relational database system), Personal Home Page (PHP), (application server and programming language), and Apache, (a widely-deployed open-source Web server). Open-source growth motivated the investigator in the development of the prototype for the medical online system. The security objective was to implement a secured access system. The access rule was: when in doubt, deny access, without sacrificing availability or performance.

## Research Questions to be Investigated

Open-source development advocates have stated that open-source tools such as PHP, MySQL, XML, and LINUX were the answer to building dynamic three-tier systems that were cost-effective, fast, and maintenance-free (Medinets, 2000). Strategic information systems continue to be a source of concern for managers. There are myths that exist concerning the impact of technology on an organization. According to Moorhead & Griffin (1989), one of the myths is the belief that expenditures of large sums of money for new technology allow an organization to reduce the skill level of employees and thus reduce overall payroll and training costs. Moorhead & Griffin also believe that, "This 'de-skilling' does occur in some situations, such as when a complex procedure and decision parameters can be programmed into a computer routine. However, more training is often required for employees to be able to run more complex systems" (p. 443).

The difficult task that management faces is the determination of the requirements of the new system (Adler, 1986). The choice of information systems must be guided by clear goals and outcomes. In researching the use of the open-source model or the traditional commercial model, managers must ascertain the effect such technological change may have on their organization. The development and implementation function must have objectives that will guide resource allocation in fulfilling organizational objectives and strategies (Adler, McDonald & MacDonald, 1992). As with any other research product, a review process of all alternatives plays an important role in enhancing and ensuring the quality of the final product. The dynamic aspect of information systems today calls for creative thought, with particular attention to quality, cost, and project control.

Open-source projects are normally structured as open teams, where the hierarchy is flattened. One team member assumes the role of leader. This person is accountable to the rest of the company for the success of the project and holds final veto power over the team's decisions (Retting, 1990). Traditional approaches are based on hierarchy that Constantine (1990) calls the 'closed paradigm.' One person has responsibility for the project, and there are supervisors under that person, each responsible for a major component of the project (Constantine, 1990).

Traditional approaches emphasize quality over process (Adler, McDonald & MacDonald, 1992). Quality and process are equally important in the open-source paradigms that are historically an open team concept. According to Retting (1990, p. 24), "Structured open teams minimize the dangers and maximize the advantages." Open teams provide creative diversity that combines abilities, shares the load, and creates built-in monitoring of the development process. Outside the open team model, there are disadvantages including overhead, communication, and personality problems. In this study, the investigator sought answers to the following questions:

1. How effective are open-source tools (PHP, MySQL, Apache Web Server, Linux Operating System) compared to commercial tools (Oracle, Windows Operating System), when used exclusively to develop a secured application for online medical collaboration?
2. How effective are open-source tools in managing session control compared to commercial tools?

3. How effective are open-source tools in managing administrative profiles compared to commercial tools?
4. How effective is an open-source MySQL database in providing access control compared to commercial databases over the Internet?
5. What is the cost advantage of open-source development techniques on the Internet compared to commercial development tools?
6. To what extent will the open-source medical online development prototype improve maintenance cost?
7. To what extent will the open-source development lower ongoing support cost?

Table 1 shows the questions, assumptions, and associated problems and goal statements. The table contains a matrix adapted from Sutor (1997) that represents the relationship between the main parts of the research and solutions adopted by the medical prototype. Hill & Jones (1992) stated that goals specify what is to be achieved and that the investigator relates them precisely to objectives and strategies to be pursued. The first column relates the questions; it is followed by the prototype's assumptions in the second column. The third and fourth columns list the related goals and problem statements discussed in the research. The final column relates the procedure used by the prototype to solve each underlying problem.

In order to determine the feasibility of the architecture, the investigator implemented an online collaborative medical prototype using exclusively open-source tools. Prototyping was the investigator's choice as a development methodology that defined and clarified the requirements for the online medical system.

Table 1: A Matrix of Research Questions, Assumptions, Problem Statements, Goal Statements and Prototype Procedures for Solving the Problem.

| Research Questions | Assumptions | Problem Statement | Goal Statement | Prototype's Procedure for Solving Problem |
|---|---|---|---|---|
| To what extent can open-source tools be used to develop secured architecture for medical collaboration? | Open-source tools help increase system's integrity, accuracy, and ease of use. | Increase vulnerability and technical reliability to protect privacy of medical data. | Determine the relevance of open-source tools to develop secured access secured, and high availability. | Three-tier database-driven prototype medical online systems. Security tools used to secure prototype from intruders. |
| To what extent can open-source manage session control, administrative profiles, and provide access control to a database over the Internet? | Security issues such as access control, and management of concurrent access. Multiple users can edit information in the same database. | Prevent denial of service attacks and security such as authentication of users, access rights, and workflow controls from free penetrating tools on the Internet. | Need to make the web-based application secured access. When in doubt, deny access. | Environmental and complimentary open source tools such as /etc/hosts.allow, Sudo, and /etc/ssh. Apache webserver provides increased security using access directives to control Internet access. |
| To what extent can open-source techniques on the Internet be implemented at a lower cost compared to commercial development tools? | Open-source tools can be implemented at a lower cost, improve extensibility, and meet legal compliance. | Take advantage of free quality open-source software and tools to reduce cost and prevent hackers. | Open-source investment is small in monetary terms. Costs will be in the way of support and consulting. No software cost. | All the tools are downloaded free of charge from open-source sites. High reliability that is viewed and worked on by thousands of developers all over the world. |
| To what extent can open-source development improve maintenance and ongoing cost? | The prototype will reduce cost as peer review improves open-source tools | Unavailability of source code of commercial applications and clear development models. | Problem resolutions and improvements are provided free by peers | Develop an ongoing cost structure by training support staff. |

According to Cusumano & Selby (1997), many companies use prototyping as well as concurrent design, build, and test activities to control iterations during system development. Prototyping explores the essential features of a proposed system through practical experimentation before implementation (Hasselbring, 2000). Prototyping facilitates better design choices early in the software development process (Zucconi, Mack, & Williams, 1990). The prototype in this study was developed using Web technology for the user interface, as a gateway for external collaboration among users, and as a messaging tool. Prototyping can help developers ask better questions of the end-users and system owners, leading to clarification and modification of existing requirements.

Database-driven development using open-source scripting languages, including Personal Home Page (PHP), Perl, and ASP, are becoming the norm in application development. Open-source scripting tools have gained vast database deployment options from commercial databases including Oracle, Sybase, and Informix. The deployment of a three-tier development process with open-source databases, scripting languages, and Web servers inside an organization is revolutionary and economical.

The investment is small in monetary terms, and the rewards are long-term, as open-source tools become core competitors to established names like Microsoft. Technology leaders including Microsoft, Netscape, Sun, Novell, and others have announced new products and technologies based on XML, Linux, and FreeBSD that promise to have a dramatic impact on the computing landscape. Internet visionaries are talking about the impact that XML will have on Internet search engines, electronic commerce, intelligent agents, seamless roaming, file systems, electronic data interchange (EDI), push technologies, software distribution, data formatting, and more (Park, 1999).

These benefits had motivated the fusion of XML, MySQL, Java Scripts and Apache Web server in the research and development team at Nova Southeastern University in the form of the Multimedia Electronic Classroom (MMECR).

According to an internal report (Microsoft Corporation, 2001), Bill Gates, Microsoft's Chairman and Chief Software Architect, expressed the vision that XML is breaking down the distinctions between Internet access, standalone applications, and computing devices of any kind. Gates indicated that XML would transform how applications talk to each other in the same way that the Web revolutionized how users talk to applications. XML benefits include the ability of businesses to collaborate to offer an unprecedented range of integrated and customized solutions on information any time, any place, and on any device (Microsoft, 2001). J.C. Levin expressed the need to collaborate as investigators and exchange intellectual resources that in turn will promote effective tools for the Internet environment and benefit both distance education and medical projects (personal communication, December 10th, 1999).

The design and implementation of the Web-based medical application allowed the separation of the business logic from the graphical user interface (GUI) and the back-end database. According to Ahn et al. (2000), the race to develop Web-enabled applications had long lacked a comprehensive security framework. The implementation included system protection and prevention of unauthorized usage that ensured oversight, enforcement, and prevention of deliberate denial of service. The explosive expansion in the use of Web browsers coupled with GUI interfaces made the three-tier application possible as well as practical. The three-tier configuration remains the most popular way to build new applications today (Maruyamaa, Tamura, & Uramoto, 1999).

Additionally, open-source tools, including XML, PHP, and JAVA, opened up new possibilities regarding the creation of new application systems by combining two or more Web applications. Vendors are increasingly using Linux, an open-source operating system, in their PCs and mobile devices (Garber, 2000). Web-based collaborative systems have recently received much attention because they support dynamic business processes over heterogeneous computing systems. The goals for this application were to establish privacy safeguards and effective mechanisms for security enforcement. Other administrative objectives were to establish an education and security awareness program with all user communities for privacy and patient confidential data.

Security issues were fundamental to the successful implementation of the three-tier online medical system. Effective security had been part of the focus in this research. Security measures were implemented for all levels of the three-tier implementation to protect information from an outsider's access without permission. XML, an open-source tool, is the most talked-about application for electronic commerce (e-commerce) because of its rich way of describing format and intelligent transport mechanisms to conduct business on the Internet.

Many companies, including banks, are moving in the XML direction for their applications development because of XML's stringent security protocols and flexibility for complex applications (Ceponkus & Hoodbhoy, 1999). It is easy to encrypt an XML stream that preserves security and flexibility using open-source scripting languages such as PHP or Java. The driving motivation for the use of open-source was to simplify security policy administration while facilitating the definition of flexible, customized policies. XML transforms how data is exchanged between applications, smart devices, and Web sites. XML-based technologies such as the Simple Object Access Model

(SOAP), which enables applications to interoperate using Internet protocols and the Universal Discovery, Description, and Integration (UDDI) that allow businesses a standard way to describe their services and connect automatically. With technologies like these, the Internet is bound to experience a revolution in Web-based services into the future (Microsoft, 2001).

## Relevance and Significance and Brief Review of the Literature

### *The advantages of open-source software and tools*

Use of core open-source tools in developing secured systems provides a multitude of benefits. These benefits include a common set of free tools that will give IT consumers' benefits in cost, security, and stability. Fraternali (1999) saw a future trend toward open-source adoption by businesses after evaluating application development tools and their adequacy for the Web. Standardized open-source tools can help promote acceptability by developers. Acceptability by the developer community in turn promotes the development of new and innovative authorization management tools by guaranteeing interoperability and portability. The open-source model has much to offer to the commercial world. It is a way businesses can build actual software by collaborating to produce a product that none of the entities may achieve alone. The open-source model also means increased security in open environments such as the Internet. The security of open-source software is possible because open-source code is in the public view, that subjects it to greater scrutiny and instant fixes to problems. Open-source software is peer-reviewed and problems are not kept secret until discovered by the client, as with most closed and proprietary software.

*Companies adopting open-source Linux*

A mature open-source code such as Linux is as bulletproof as software is expected to become. Linux is gaining credibility in the corporate sector faster than commercial operating systems. Firms that have made a major commitment to Linux include Home Depot, Cisco, and IBM. IBM is spending over $1 billion on Linux software (Leavitt, 2001). In 2001, both Silicon Graphics and Sun Microsystems had embraced the open-source model by deciding to expose a portion of their respective Unix operating systems (Shankland, 1999). Strings of other large companies, ranging from E-Trade Inc. to DreamWorks SKG, announced deployment plans for Linux (Galli, 2002).

Table 2 shows the companies that adopted the Linux open-source operating system. According to Galli (2002), the companies listed in Table 2 replaced their commercial operating systems with the open-source Linux operating system. The table shows the names of the companies, the projected cost savings anticipated, and some of the reasons cited as influencing the decision to use the open-source Linux operating system.

Linux provides advantages in terms of price and performance (Galli, 2002). Boscov's, a store chain that operates 35 department stores, had consolidated by porting its Windows NT server's farm to a single IBM z900 server running Linux. The company plans to move its current print and file servers to Linux virtual servers and its e-commerce and database applications to a Linux mainframe over the next few months. According to Galli (2002), the President of Pixar Technology, Darwyn Peachey, believes that Linux offers the company competitive advantages, including greater performance, higher productivity, and immense creativity. "This will be the world's first Linux IA-64 render farm, giving DreamWorks artists the ability to create larger, more complex images," said Jeffrey Katzenberg (Galli, 2002, p.1).

Table 2: Recent Linux Deployment Matrix (2002)

| Company | Estimated | Reason cited for Deployment |
|---|---|---|
| Bosov's (Department Store chain) | N/A | Consolidated a Windows NT server farm on a single z900 running Linux to reduce cost and improve services. |
| E-Trade (Online Trading) | $65 million | Replaced Solaris servers in favor of Linux-based servers to save cost and improve online business |
| Salomon Smith Barney (Financial) | N/A | Replaced mainframe operating system with SuSe 7.0; plans to port virtualization application to Linux to improve business processes. |
| Pixar Animation Studios | N/A | Ported 500 million lines of code to Linux and plans to move from SGI workstations to Linux. |
| DreamWorks Studios | N/A | Moved to Linux for the rendering and animation of its digital movies; collaborating with Hewlett Packard (HP) to optimize its animation application for the IA-64 architecture. |

According to Shankland (2001), Jordan Hubbard, one of the co-founders of FreeBSD open-source operating system, announced that Apple Computers hired him to work on the open-source underpinning of Apple's OS-X operating systems. Jordan Hubbard will be responsible for leveraging BSD technology as part of the Mac OS-X in partnership with the open-source community (Shankland, 2001). In addition to the high adoption rate of open-source tools, there is a growing abundance of skills in open-source tools like Linux coming out of U.S. colleges and universities (Babcock, 2000). The open-source community is helping spawn software development collaboration on the Internet.

*The growing popularity of the open-source database, MySQL*

A database-driven Web application will significantly benefit from open-source adoption by developers. MySQL is one database engine that will compete favorably with databases such as Oracle, Postgres, and Sybase. MySQL has many features to offer. Compared with other database, MySQL has great performance, support, and features such as SQL conformance. MySQL is a high-performance but relatively a simple database system, and is easier to administer than larger databases like Oracle (DuBois, 2000). According to DuBois, cost, query language support (SQL) and ODBC (Open Database Connectivity), security, and open-source availability make MySQL the most used database on the Internet (DuBois, 2000).

The objectives of this research included the protection of the database against intentional or unintentional threats using open-source controls. A database represents an essential corporate resource that should be properly secured using appropriate controls (Connolly, Begg, & Strachan, 1999). The authors emphasized that database security had to be considered in relation to theft, fraud, confidentiality, privacy, integrity and availability. In this study, the investigator assumed that MySQL, an open-source database, would satisfy all the security requirements of Connolly, Begg, & Strachan (1999). MySQL's vast deployment on the Internet is based on its excellent security characteristics and low cost. Shared applications are imperative to geographically dispersed collaborators on the Internet.

*Advantages of online open-source collaborations*

Today's health care industry is associated with various economic risks. Healthcare planners are calling for budget cuts through reduced administrative costs while simultaneously retaining high-quality patient care. As in healthcare and most industries, the U.S. is an exception among industrialized countries. Trillions of dollars are expended in the industry, yet the U.S. is without a national health insurance system (Huston & Huston, 2000). The collaborative medical online system will help to increase productivity by cost containment through effective collaboration between participating providers and insurers.

Online collaboration will provide health insurance companies the means to share and communicate with their numerous providers, including geographically dispersed medical offices. It will eliminate major printing, mailing, and static data common with today's healthcare services. The use of mostly open-source software tools will further reduce the cost in system development and system maintenance (Qusterhout, 1999). Open-source software offers an opportunity to prevent the risk of inadequate robustness associated with proprietary software (Garber, 2000; Newmann, 1999). When commercial systems are not adequately robust and secure, designers may consider open-source components to build robust systems (Neumann, 1999). Open-source is the most significant advancement in computing now shaping the world's economy and the little understood 'hacker culture'. Open-source has been argued as a superior methodology for the development of end-user software that increases economic assets of companies (O'Reilly, 1999).

E-Trade Corporation, an online brokerage firm, has demonstrated that a well-designed Internet site could have a positive effect on profitability. The Internet provides an excellent opportunity to reach customers (Spiegel, 1999). The Internet can be used as an inexpensive yet powerful alternative to other forms of communication. As more businesses engage in globalization, inter-organizational collaborative computing grows in importance.

With limited homogeneous computing environments in participating organizations, heterogeneity and Internet-based technology are prevalent in inter-organizational collaborative computing environments (Kang, Park, & Froscher, 2001). According to Barkley, Kuhn, Rosenthal, Skall, & Cincotta (1998), a plethora of corporate information (e.g., procedures, training materials, directories, forms, and databases) can be converted to electronic form and accessed through the Internet. With a single source of information, cost of maintenance is significantly reduced, while greatly simplifying the task of ensuring currency. Thus, an objective of enterprise computing, according to Barkley et al. (1998), is the creation of a company-wide system irrespective of the underlying information technology infrastructure. Companies are seizing the Internet as a quick way to streamline and transform their organization's bottom line. Most organizations are becoming aware of the need to share information contained in their databases over the Internet. The reasons for the need include better communication between employees, mobile executives' instant access to corporate data, and the movement towards e-commerce (Bradley, 1998).

Most of the difficult tasks include organizing and delivering information on the Web, as well as exchanging information among incompatible systems and applications (Goulde, 1999). Open-source tools like Linux and XML are technologies that will help solve most incompatibility problems persistent in information technology (IT) today. The commercial Unix community had never anticipated that Linux would take market share away from Microsoft since Windows 3.0 came on the scene in the mid-1980s and swept away the desktop space. Open-source Linux now appears to be doing better than holding its own against Windows on key servers inside emerging online companies (Babcock, 2000). According to its proponents, open-source style software development has the capacity to compete successfully, and perhaps in many cases displace traditional commercial development methods (Mockus, Fielding, & Herbsleb, 2000).

The Web provides the minimum for setting up enterprises by enabling identification of shared goals that are supported and coordinated via email and other enabling tools for collaboration. Humans have a natural propensity to collaborate with one another, and this collaboration is enhanced by new opportunities like the Web (Fielding, Whitehead, Anderson, Bolcer, Oreizy, & Taylor, 1998). One concern of businesses using Internet messaging is security. The Internet is a public network, and there had been no adequate protection against attacks such as eavesdropping and forgery. Government computer experts recently reported a compromise in computer security in which hackers stole large amounts of data from Department of Defense computers. The incident highlights the new role the U.S. military and intelligence agencies are playing against unseen intruders (Bridis, 2001).

## Barriers and Issues

This research is about the implementation of a three-tier open-source prototype for an integrated medical system on the Internet. Security and integrity is essential to minimizing the possibility of unauthorized access and modification or destruction of medical information involving providers (clinics, hospitals, etc.,) and carriers or Health Maintenance Organizations (HMOs). The investigator assumes that improved communication security, firewalls, and router configuration are adequately protected from attacks perpetrated through the modification of scripts and environmental programs. Medical online systems have not been feasible because of the need for strict security for medical data. Other barriers include system resources such as tools, cost, and data inaccessibility because of medical information's distribution across multiple geographical sites. Completely securing an online application against unauthorized access is extremely difficult (Bashir, Serafini, & Wall, 2001). Internet security is a multidimensional framework to help safeguard against attacks from outside as well as within an organization (Phillips & Spakovsky, 2001). Some of these dimensions include privacy, physical access restrictions, application availability, network confidentiality, content integrity, and access policy (Bashir, Serafini, & Wall, 2001). Online attacks such as virus propagation and deliberate denial-of-service are common on the Web (Weinstein & Neumann, 2000).

The scope of this implementation included essential components in the prototype to protect the contents of data and related environmental variables used to improve access security. Risk assessment was used to measure the impact of threats on the medical system and how to control these threats. Implementing security after a risk assessment may involve measures to reduce a server's vulnerability, such as allowing another server to take over if one server goes down (Cooper, Goggans, Halvey, Hughes, Morgan, Siyan, Stallings, & Stephenson, 1995). Internal attacks should be an organization's number one information security objective. The issues of MySQL security, object databases, addressing the separation of duties, prototyping shortcomings, accidental disclosures, physical security, network security, insider curiosity and subordination, healthcare computing environment and privacy, and default and guessable passwords were the barriers examined as important challenges.

*Performance and security compromises*

Security consideration may constrain the desire to combine MySQL tables. The main reason is that combined tables allow user data in the same table. It is advisable to allow access to each table by each user by means of table-level privileges. MySQL has no provision for restricting access to particular rows to a given user like an oracle database (DuBois, 2000). In some cases, merging database tables may compromise access security. Older MySQL databases had internal limits on table size of 4GB prior to MySQL version 3.23. Newer MySQL versions have approximately 9 million terabytes limits on table sizes, as opposed to 64GB for commercial databases on IBM's AIX 4.2 operating system. Han (2001) concluded that many large corporations have passed over MySQL as a high-performance database because of perceived views of *transactional* and *high availability.* A transaction is a grouping of SQL statements executing as a unit; transactions either succeed or fail as a group. Transactions allow users to complete their data inserts before new users are handed control. MySQL uses metadata, which does not rely on transactions and restricts end users' direct request to the database. An application, such as a Java engine, uses metadata to manipulate data objects and manage concurrences. The application also maintains data integrity because of the many databases that *read* rather than *write.* High availability is another MySQL weakness. The solution of metadata issues relative to transactions is to use third party tools such as mission critical Linux high-availability for MySQL clustering (Han, 2001; DuBois, 2000).

Developing a robust security model for object relational databases is still problematic. The question of object inheritance and standardization remains the problem of adequately securing an object-based computing environment. For example, when an object is created, does it inherit the access levels of its class? Can the creator or owner of a class have access to all objects created from the class? How should access rules generally propagate among objects, classes, and super-classes? Blaha & Premerlani (1998) suggest the definition of privileges for prototypical users such as clerks, managers, and system administrators. This way, each actual user then receives permission from a prototypical user.

*Separation of duties with access control*

Another security issue is the concept of *separation of duties*. The concept stresses the importance of *separations of duties* as it relates to general security implementation (Mayfield, Roskos, Welke, & Boone, 1991). Mayfield, et al. suggests that any critical task such as deploying nuclear missiles that is performed by a system must involve two or more people. The intent is to make unauthorized access more difficult and to allow some level of checks and balances. For example, one person may not be allowed to submit a checks request, authorize the funds, and withdraw the funds from a corporate account.

*Shortcoming of prototyping*

The use of prototyping in software design can lead to some problems. The problems include a less coherent design as compared to some specification-based approach, particularly the traditional waterfall design model. Other prototyping shortcomings are poor (or nonexistent) documentation and the tendency to deliver the prototype as a finished product because of schedule pressures to deliver the system (Zucconi, Mack, & Williams, 1990).

*Accidental disclosures of information*

Innocent mistakes may cause unintentional disclosure of information to outsiders. Examples include conversations that may be heard between privileged personnel in an office corridor, on the telephone, or in the elevator. Clerical staff may notice and retrieve privileged information for an acquaintance or family member. Privileged information may be left on a computer screen to be abused by a stranger, or a provider may misaddress e-mail and other digital messages unintentionally. Johnson (1997) believed that with electronic networks, there is no loss of value in the process of reproduction. A copied program or data set is perfectly usable, and the reproduction can be such that there is no evidence that copying had been done.

*Physical access to computing environments*

The weakest link in any security policy involves people who are authorized to use a computing environment. The objective of computing security using technology is possible, yet a complex task. Physical access issues arise from the people trusted with the computing environment and not in the technology. Once a person with malicious intent has gained physical access to a computer, everything from booting the system in single-user mode to physically removing the system becomes possible (Mann &Mitchell, 2000).

*Network security*

Network attacks have become more serious (Rieken & Weiman, 1992) and more difficult to defend against. Networks are commonly implemented over electrically based media, such as 10/100 base T, radio frequency (RF) communications, microwave technology, and satellite communication.

All of these forms of communication may be intercepted through various tapping techniques such as network scanning (Barret, 1996).

*Insider curiosity and subordination*

Users may abuse their record access privileges out of curiosity or for their own information purposes that would include getting unauthorized access to the medical records of family and friends. In some cases, users may also want information about celebrities or about the possibility of sexually transmitted diseases about someone that they are dating. Technological obstacles such as access profiles and computer authentication that ensure that users access only information for which they have a bona fide need and right to know can be equally effective (Rindfleisch, 1997).

*Healthcare computing environment and privacy*

The healthcare industry has undergone profound changes in business practices. Graphic images, annotations, and other elements that convey medical data are now transmitted over high-speed networks at tremendous infrastructure cost. Deliveries of multimedia data to many users present several issues. In general, when data is large, access and delivery can easily be a bottleneck (Rindfleisch, 1997; Gordon & Loeb, 2001). For example, delivery of video to many medical centers at once in a medical emergency such as a biological attack may be problematic. The possibility of each transmission remaining in force and delivered at the standard rate may not hold. Delivery issues may cause the video to undergo a time distortion, although certain kinds of video/audio can support different kinds of delays or delivery failure, quality of service can equally be affected. In addition, issues arise where physicians need to draw upon many different kinds of data in the course of their work. Medical records about a single patient may exist at many different hospitals, medical offices, and other insurance offices not part of the online collaboration. The data of a patient may have to be gathered from these outside sources. Data about procedures, drugs, and other aids to treating the patient are, in principle, available through multiple systems. These possible data sources, such as records of diagnostic tests, insurance, and other billing data may have a major impact on cost, and possibly on quality of care. Finally, an access control to preserve the confidentiality of medical records from all outside sources is an issue with legal consequences for the medical industry (Gordon & Loeb, 2001; Huston, 2001).

The Internet is a collection of databases and other information sources that are connected through a network owned by different participants. Healthcare systems as participants of the Internet may encounter special problems such as participants refusing to accept connections or encountering systems with specific capabilities (Gordon & Loeb, 2001). For example, a server may insist on remuneration in return for performing a specific service such as an inquiry on medical symptoms from an online service provider. Medical processes such as x-rays may involve both machine-based steps and human-based steps, where a person is required to intervene. For example, the task of entering patient x-ray and physician annotations involves a sequence of steps including scanning and data entry. Some events may involve long delays, due to issues such as the office clerk being on vacation and a new backup clerk being at lunch, affecting workflow and receipt of timely data. Rindfleisch (1997) asserted that technology could do little to ensure that people receiving information will handle it according to standards.

*Default and guessable passwords*

There may be considerable innovation on the Internet using software with other devices over time to protect privacy of users and information. A few software and hardware vulnerabilities account for the majority of successful attacks because attackers are opportunistic, using the easiest and most convenient routes (Mann & Mitchell, 2000). Some systems come with *demo* or *guest* accounts with no passwords or with widely known default passwords. Problems arise when maintenance accounts with no passwords or accounts with default passwords are left unchanged. In addition, busy system administrators often select system passwords that are easily guessable ("love," "money," and "wizard" are common) or the use of a blank password. Default passwords provide effortless access for attackers. Many attackers try default passwords and then try to guess passwords prior to using methods that are more sophisticated. Compromised user accounts wind up with attackers inside the firewall and inside the target machine. Once inside, most attackers can use widely accessible exploits to gain root or administrator access (Felton & Schneider, 2000; Mann & Mitchell, 2000).

Ricoeur (1974) states that in the fabric of our innermost being lay the freedom to deliberate, decide, and act. People are rational beings and they can choose to do either good deeds or bad ones. It is each individual's decision as to social and personal responsibilities.

## Delimitations and Limitations

This study was conducted under financial and time constraints; therefore, the investigator encountered some limitations and delimitations. The limitations include the cost of implementing security that involved compromises between performance and proactive monitoring. Other financial constraints included the cost of resources for development, execution, and maintenance (CPU, memory, disk space, support staff size, and utilization).

A delimitation was generated by the need to become familiar with object orientation methods. In reality, object-oriented design and programming are difficult to learn. According to Ledgard (2001, p. 126), "It may take the average programmer two years to master object-oriented programming. In general, the concept of black box can be confusing." An object is like a black box and has state, behavior, and identity. The structure and behavior of an object are defined within a common class (Blaha & Premerlani, 1998). Object orientation facilitates the concept of inheritance that is hard to anticipate from procedural programming thinking. Inheritance provides a natural classification for objects and allows for the commonality of objects to be explicitly taken advantage of in modeling and constructing object-oriented systems. Objects allow the means to use complex concepts, classifications, and generalizations to understand and deal with real world problems (Blaha, 2001). To re-use a class, the programmer must understand the inner workings of the class. In a sense, a programmer with procedural knowledge has to understand the notion of abstraction to succeed (Ledgard, 2001).

This study was limited in that the author did not intend to obtain open-source preferences from other disciplines. The study was conducted in the computing discipline. There may have been some bias toward certain products from the open-source community that could have been considered anti-capitalist because of the dominance of the global information economy by a few American companies. As stated by Armour (2002), as a result of college preparation and the cerebral nature of system development, most software engineers get little or no exposure to team concepts compared to other disciplines like business administration. There are some disagreements that the average social need for software developers is significantly below the “average” for the population at-large. Armour (2002), and Daniel & Zawacki (1980) disagree with some aspects of the social view of software developers. The investigator believes there is no significant social difference between developers and others in the general population. More research is needed to determine whether the anonymous nature of online collaboration may account for the success of most open-source projects, or if the new software developer may naturally, within the context of the development environment, become social.

## Assumptions

In this study, the investigator made several assumptions based on experience and as a proponent of the open-source model. Open-source is producing many contributions in the field of software development in addition to impacting the information technology market. The assumptions are:

1. Top-down and the bottom-up software design methodologies that involved centralized management and control on the development process failed to produce quality software. Clearly defined design and rigorous management are not instrumental in open-source development and coding projects. Traditional models may support clearly defined design and rigid control; the traditional model is dispelled in practice by many open-source projects that have succeeded even without a clean initial design and formal management control process.

2. Commercial software companies prevent access to their source code because they make profits from support, upgrades, and other income sources that tie in customers to their products. However, the profit assumption is challenged by the success of several open-source projects. Open-source software provides reduced risk of commercial software profit motives as improvements can be tracked iteratively online, thereby reducing support and upgrade costs.
3. Large numbers of developers spread around the world have collaborated to build reliable and high quality software like the Apache Web server. A small open-source project can muster more brains to improve software than most development shops can possibly afford.
4. The open-source paradigm and object-oriented methodology provide general disciplines that address both design and user collaboration for using prototyping to produce better software applications.
5. Building broad community involvement allows improved reference points for efficient software engineering. Boehm & Basili (2000) agreed that both the strength and weakness of open-source software is that it is abstract by nature and apparently limitless and flexible.
6. The motivation of the open-source community can be understood through Murray's manifest theory. Achievement, autonomy, and understanding are the set of needs that motivates members of the open-source community.

The investigator supported the open-source paradigm by developing a multi-tier online prototype that integrated the Hypertext Transfer Protocol (HTTP) and a client/server application for better performance, scalability, reliability, and security. The investigator assumed that improved communication security, firewalls, and router configuration were adequately protected from attacks perpetrated through the modification of scripts and environmental programs.

## Definition of Terms

For the purpose of this study, the following definitions of terms are used.

**Abstraction:** Abstraction is the ability to focus on essential aspects of an application while ignoring design details. Abstractions give one the ability to preserve design freedom until later stages of development when more informed decisions could be made (Blaha & Premerlani, 1998).

**Active Server Page (ASP):** ASP is a file containing markup language, server script, and client script that are processed by the Active Server Processor in Microsoft Internet Information Server (DuBois, 2000).

**American National Standards Institute (ANSI):** ANSI is the registration authority in the U.S. for organization names under the global registration process established by ISO. The registration service provides an unambiguous organization identifier (Deitel & Deitel, 1999).

**Apache Web Server:** The Apache (httpd) server is a powerful, flexible, HTTP/1.1 compliant Web server that implements the latest protocols, including HTTP/1.1 (RFC2616). An Apache is a highly configurable and extensible server. Writing "modules" using the Apache module API provide a full source code that comes with nonrestrictive license and can be customized by third-party modules. The server runs on Windows NT/9x, Netware 5.x, OS/2, and most versions of Unix, as well as several other operating systems. Apache is an open-source server actively being developed and encourages user feedback through new ideas, bug reports, and patches. It implements many frequently requested features, including: DBM databases for authentication that allows the user to easily set up password-protected pages with enormous numbers of authorized users without bogging down the server (DuBois, 2000).

**Bandwidth:** Bandwidth refers to the range of frequencies available for signaling. The difference expressed in Hertz between the lowest and the highest frequencies of a band (Davis & McGuffin, 1995).

**Bell-LaPadula:** Bell-LaPadula is a model that aims to enforce the principles of confidentiality and describes the basic functions of a multilevel-secure system (Bell-LaPadula, 1973).

**Berkeley Internet Name Domain (BIND):** The Berkeley Internet Name Domain (BIND) implements an Internet name server for BSD-derived operating systems. The BIND consists of a name server (or ``daemon") and a resolver library. A name server is a network service that enables clients to name resources or objects and share this information with other objects in the network. This in effect is a distributed data base system for objects in a computer network. The BIND server runs in the background, servicing queries on a well-known network port. The standard port for UDP and TCP is specified in /etc/services (DuBois, 2000). The resolve is a set of routines residing in a system library that provide the interface that programs can use to access the domain name services. The system administrator can configure the system to use BIND as a replacement to the older host table lookup of information in the network hosts file /etc/hosts. The default configuration for BSD uses BIND.

**Broadband:** Broadband is a transmission system where signals are encoded and modulated into different frequencies and then transmitted simultaneously with other signals (Davis & McGuffin, 1995).

**Brooks' Law:** Brooks' law means that adding manpower late in a software project assumes that the result of the expected advantage from splitting development work among N programmers is O (N); that is, proportional to N, but the complexity and communications cost associated with coordinating and then merging their work is O ($N^2$); that is, proportional to the square of N (Drummond, 2000).

**Client/Server Computing:** Client/server computing is a network environment designed with a processor or computer designated as the server providing services to other client processors or computers. Client/server computing is the logical extension of modular programming. The modular programming fundamental assumption is the separation of a large piece of software into its constituent parts ("modules") and creates the possibility for easier development and better maintainability. Client/server computing takes this a step further by recognizing that those modules need not all be executed within the same memory space. With this architecture, the calling module becomes the "client", and the called module becomes the "server" (Davis & McGuffin, 1995; Taylor, 2002).

**Computing Environment:** Computing environment relates computers, networks, and their collective interactions at a particular time. The functional requirements of a computing environment may include factors such as reliability, integrity, and confidentiality (Mann & Mitchell, 2000).

**Confidentiality:** Confidentiality is the concept of making computing resources, especially the data contained therein, available only to authorized personnel (Mann & Mitchell, 2000).

**Cryptography:** Cryptography is the transformation of plain text into coded form (encryption) or from coded form to plain text (decryption). Cryptography is the science of transforming information that can be read (in plain text) into information that cannot read. In this process, information is coded (encryption) to stop information from being read or altered by anyone but the intended recipient. It may be intercepted, but it will not be intelligible to someone without the ability to decode (decryption) the message. Encryption and decryption require a mathematical formula or "algorithm" and a key to convert data between readable and encoded formats. A key is a unique number that is combined with the plain text to produce the encrypted message or the digital signature (Davis & McGuffin, 1995; Mann & Mitchell, 2000).

**Database Management System (DBMS):** Database management system is a set of programs used to define, administer, and process the database and its application (Kroenke, 2002).

**Database Server:** A database server is a backend processor that manages the database and fulfills database requests in a client/server environment (Davis & McGuffin, 1995).

**Data Integrity:** Data integrity is verified correspondence between the computer representation of information and the real-world events that the information represents. The condition of being whole, compete, accurate, and timely (Davis & McGuffin, 1995).

**Data Security:** Data security is the result achieved through implementing measures to protect data against unauthorized events leading to unintentional or intentional modification, destruction, or disclosure of data (Davis & McGuffin, 1995).

**Digital Subscriber Line (DSL):** Digital subscriber line is a transmission access system using copper (telephone). It is a network from voice-band modems to Digital Subscriber Line (DSL) technologies (Peden & Young, 2001).

**Domain Name Server (DNS):** A DNS directory service consists of DNS data, DNS servers, and Internet protocols for fetching data from the servers. The billions of resource records in the DNS directory are split into millions of files called zones. Zones are kept on authoritative servers distributed all over the Internet, which answer queries according to the DNS network protocols. In contrast, caching servers simply query the authoritative servers and cache any replies. Most servers are authoritative for some zones and perform a caching function for all other DNS information (Medinets, 2000). Most DNS servers are authoritative for just a few zones, but larger servers are authoritative for many zones.

**Eavesdropping:** Eavesdropping is an unauthorized interception of data transmission. Modern communications systems are virtually transparent to the advanced interception equipment that can be used to listen in. Eavesdropping is and will remain a security problem. Some systems even lend themselves to a dual role as a national interceptions network (Davis & McGuffin, 1995; Mann & Mitchell, 2000).

**Eclipse Project:** The Eclipse Project is an open source project of eclipse.org, overseen by a Project Management Committee (PMC) and project leaders. The work is done in subprojects working against a repository. The Eclipse Project Charter describes the organization of the project, roles and responsibilities of the participants, and top-level development process for the project (Berlind, 2002).

**Encapsulation:** Encapsulation is the separation of external specification from internal implementation (Blaha and Premerlani, 1998). Each object is somewhat self-contained and contains data and code to manipulate the data code and the data to support that code. Each object can be seen as a "black box" and code can be updated as long as the interface remains the same.

**Extensible Markup Language (XML):** XML is an incredibly powerful system for managing information. Internet visionaries are talking about the impact XML will have on Internet search engines, electronic commerce, intelligent agents, seamless roaming, file systems, electronic data interchange (EDI), push technologies, software distribution, data formatting, and more (Park, 1999).

**FreeBSD:** FreeBSD is a state-of-the-art operating system for personal computers based on the Intel CPU architecture. FreeBSD provides many advanced features previously available only on much more expensive computers. These features include preemptive multitasking with dynamic priority adjustment to ensure smooth and fair sharing of the computer between applications and users. Multi-user access means that many people can use a FreeBSD system simultaneously for a variety of applications or purposes. Complete TCP/IP networking including SLIP, PPP, NFS and NIS support. A FreeBSD machine can inter-operate easily with other systems as well act as an enterprise server, providing vital functions such as NFS (remote file access) and e-mail services or putting an organization on the Internet with WWW, FTP, routing and firewall (security) services (Urban & Tiemann, 2002).

**Hacker:** Hacker in a positive mode is a computer enthusiast. A computer enthusiast is a person who enjoys an intellectual challenge of creatively overcoming and circumventing limitations of computing systems (Davis & McGuffin, 1995). The negative hacker is a malicious meddler who tries to discover sensitive information by poking around. A hacker may be a `password hacker' or a `network hacker'.

**Health Maintenance Organizations (HMO):** HMOs are managed care companies or insurance companies that employers pay a premium to for employees' medical coverage. Participating hospitals give the customer (patient) a warranty with the purchase of health care (Gorden & Loeb, 2001).

**Internet Information Server (IIS):** IIS is a Web server; that is, it is a collection of software programs designed to service requests for information and other resources from clients on the Internet, World Wide Web, or organizational intranets. In a broader sense, IIS is a comprehensive Web server and Web-publishing system designed especially for use with the Microsoft Windows NT Server operating system (Microsoft, 1998).

**IBM DB2 Universal Database:** IBM's DB2 Universal Database family of products consists of database servers and a suite of related products. DB2 is available on many hardware and operating system platforms, ranging from mainframes and large servers to workstations. The first DB2 product was released in 1984 on the IBM mainframe platform (Silberschatz, Korth & Sudarshan, 2002).

**Integrity:** Integrity refers to the correctness of data contained in a particular computing environment. The concept incorporates application data (password files, routing tables, DNS information, etc.) Integrity compromises occur from software bugs, accidental introduction of errors, and deliberate modifications of data (Mann & Mitchell, 2000).

**Internet Service Provider (ISP):** An ISP (Internet Service Provider) is a company that provides third party access to the Internet. Customers simply use their modem to connect to the ISP, which then links them to the Internet automatically. Although the prices and facilities of ISPs differ, they all offer some standard basic services such as 24/7 Internet access, a unique e-mail address and storage space for a Web site, and basic software programs for browsing the Internet. Some ISPs also referred to as on-line information providers, provide extra services such as access to databases of business information (Microsoft, 1998).

**Institute of Electrical and Electronics Engineers (IEEE):** The IEEE (Eye-triple-E) is a non-profit, technical professional association of more than 377,000 individual members in 150 countries. The full name is the Institute of Electrical and Electronics Engineers, Inc., although the organization is most popularly known and referred to by the letters I-E-E-E. Through its members, the IEEE is a leading authority in technical areas ranging from computer engineering, biomedical technology, and telecommunications, to electric power, aerospace, and consumer electronics, among others. Through its technical publishing, conferences and consensus-based standards activities, the IEEE publishes 30 percent of the world's literature on electrical engineering, computers, and control technology. IEEE holds annually more than 300 major conferences, and has more than 860 active standards with 700 under development (Davis & McGuffin, 1995).

**Java Database Connectivity (JDBC):** JDBC is a Java API for executing SQL statements. (As a point of interest, JDBC is a trademarked name and is not an acronym; nevertheless, JDBC is often thought of as standing for ``Java Database Connectivity".) It consists of a set of classes and interfaces written in the Java programming language. JDBC provides a standard API for tool/database developers and makes it possible to write database applications using a pure Java API (Deitel & Deitel, 1999).

**Kernel:** The kernel is the heart of any operating system. The kernel performs low-level tasks such as memory allocation, process management, and communication with hardware. It serves as the negotiator between programs and the hardware of a computer system. Kernels are some of the most difficult and complex of all types of computer programs (Drummond, 2000).

**Linux Operating System:** Linux is a free. It is licensed under the General Public License (GPL) to promote the publication of free software. The operating system is based heavily on the POSIX and UNIX API's. It supports both 32-bit and 64-bit hardware and provides a stable multi-user Internet-ready operating system. Linux, itself, is not Unix, although many people call it that and one would be hard-pressed to tell the difference. This is because the Unix trademark is specific to systems that meet a complex set of X/Open standards and has a cost (DuBois, 2000).

**Microsoft SQL Server:** Microsoft SQL server is a complete data management package including relational database server, full text indexing and search, XML data import and export, distributed and heterogeneous data integration, data transformation engine, and more that makes development more manageable (Blaha & Premerlani, 1998).

**Metadata:** Metadata describes the structure of data in a database stored in the data dictionary. Metadata are used to describe tables, columns, constraints, and indexes (Kroenke, 2001).

**MySQL:** MySQL is a very fast, multi-threaded, multi-user, and robust SQL (Structured Query Language) database server. MySQL is open-source Software. The database can be used and modified by anyone. Anybody can download MySQL from the Internet and use it without paying anything. Those so inclined can study the source code and change it to fit their needs (DuBois, 2000)

**Multimedia Electronic Classroom (MMECR):** MMECR is the collaborative online environment (tool) for investigators and students of Nova Southeastern University targeting distance learning, developed through a collaborative effort by Dr. J. Levin. MMECR is a valuable teaching tool for online collaboration. It is an electronic discussion forum that is playing an important role as a supplement to traditional classrooms (Brown, Ash & Rutherford, 1993).

**Multi-User Domains (MUD):** MUD is software that accepts user connections across a network. The user connecting through telephone, Internet, and other broadband connections provides users access to a shared database of virtual rooms, exits, and other objects. Each user browses and manipulates the underlying database from the virtual rooms, seeing only those objects that are common in the same room and moving to other rooms via the exits that connect them (Curtis, 1992; Hampel & Keil-Slawik, 2001).

**Multi-User Object-Oriented Environment (MOO):** MOOs are some form of a social MUD for virtual online meetings of users. MOO architectures are database-oriented and based on interacting objects communicating on specific events. Some MOOs are used as technical platforms that may integrate into an existing learning environment. MOOs are also used for teaching purposes, designed to enable users to work actively and with material (Hampel & Keil-Slawik, 2001).

**Open-source:** The open-source model has a lot to offer the business world. It is a way to build open standards as actual software, rather than paper documents. It is a way where many companies and individuals can collaborate on a product that none of them could achieve alone. The open-source model also means increased security because code in the public view will be exposed to extreme scrutiny, with problems being found and fixed instead of being kept secret until the wrong person discovers them. And last but not least, it is a way that the little guys can get together and have a good chance at beating a monopoly (Raymond, 1999).

**Object:** There are many definitions of an object, such as found in (Booch, 1991). An object has state, behavior, and identity. The structure and behavior of similar objects are defined in their common class; the terms instance and object are interchangeable.

**Object Orientation (OO):** An Object Orientation is a strategy for organizing systems as collections of interacting objects that combine data and behavior. Object-oriented modeling permeates the life cycle and applies to implementation (Blaha and Premerlani, 1998).

**Open Database Connectivity (ODBC):** ODBC is a standard interface by which application programs can access and process SQL databases in a DBMS-independent manner (Kroenke, 2001).

**Object Management Group (OMG):** OMG is a group that is developing standards for object-oriented computing (Blaha & Premerlani, 1998)

**Practical Extraction and Report Language (PERL):** PERL is an interpreted programming language originally developed by Larry Wall, for UNIX operating systems. PERL interpreters are now available for Windows and other systems. PERL is mostly used for server programs on the Internet and intranets (Medinets, 2000).

**Personal Home Page (PHP):** PHP is a popular open-source HTML-embedded scripting language. Much of its syntax is borrowed from C, Java, and Perl, with a couple of unique PHP-specific features added. The goal of the language is to allow Web developers to write dynamically generated pages quickly (Medinets, 2000).

**PostgreSQL:** The original Postgres code, from which POSTGRESQL is derived, was the effort of many graduate students, undergraduate students, and staff programmers working under the direction of Professor Michael Stonebraker at the University of California, Berkeley. When SQL functionality was added in 1995, its name was changed to Postgres95. The name was changed at the end of 1996 to POSTGRESQL (Danesh, 1999).

**Reliability:** A computing environment is considered reliable if it behaves in the way in which it is configured at all times. Two factors contribute to the reduction in the reliability of a system. Some factors include unscheduled downtime and Denial of Services (DoS) attack (Mann & Mitchell, 2000).

**Simple Object Access Protocol (SOAP):** SOAP is a protocol for transmitting procedure calls and XML documents using HTTP (Kroenke, 2001).

**Structured Query Language (SQL):** SQL is a powerful database language used for accessing the data of a relational database. (Kroenke, 2001).

**Telemedicine:** Telemedicine allows health care professionals to use medical devices in the evaluation, diagnosis, and treatment of patients in other locations. These devices are enhanced through the use of telecommunications technology, network computing, and video-conferencing systems. Specialized application software, data storage devices, database management software, and medical devices capable of electronic data collection storage, and transmission are all key components of the Telemedicine infrastructure (Huston & Huston, 2000).

**Unified Modeling Language (UML):** UML is a language used to specify, visualize and document the artifacts of an object-oriented system under development. It represents the unification of the Booch, OMT, and Objectory notations. Booch, Rumbaugh, and Jacobson developed it for modeling applications. (Blaha, 2001).

**Use Cases:** A use case is a technique used to describe what a new system should do or what an existing system already does from the user viewpoint. An important aim of use cases is to show the functional requirements of a system (Blaha, 2001).

**World Wide Web Consortium (W3C):** The World Wide Web Consortium (W3C) develops interoperable technologies (specifications, guidelines, software, and tools) to lead the Web to its full potential as a forum for information, commerce, communication, and collective understanding. W3C is jointly working in the area of privacy with a variety of member organizations and institutions and a number of invited experts from the US, Europe and Asia (Walmsley, 2001).

## Summary

In this chapter, the investigator presented an introduction to the scope of the study and discussed the significance and problems of software development using exclusively open-source tools. The investigator presented arguments and explored design issues relevant to the Internet's collaborative environments. The investigator also reflected on the changing popularity of the Internet and its contribution to the open-source community. The chapter emphasized the open-source model and its exploding adoption by commercial organizations, creating wider acceptance and a support base for open-source applications. The investigator showed how the effectiveness of communication and online collaboration could be enhanced when groups' activities are coordinated. It is also noted that, without coordination, a collective team may engage in conflicting actions.

The chapter examined the different security and application development strategies for an online medical collaboration. The problem statement reviewed the security issues confronting sensitive information on the Internet. The security problems included the openness of the Internet and the availability of free tools and instructions on how to compromise systems. The waterfall software development approach was reviewed and found inappropriate for applications with a level of uncertainties including online applications. Waterfall approach is not recommended for applications with substantial uncertainty in the requirements (Blaha, 2001). The waterfall model is great for applications that are well understood and the risks of changing requirements are low (Boehm, 1988). The goals were to select a software development methodology that addressed the uncertainties of online application requirements and the ability to use open-source software and tools. The research questions examined in chapter 1 included the effectiveness of open-source tools for online activities including security, cost, and session control. The advantages and significance of the open-source model in promoting common set of free tools, and benefits in cost, security, and software stability were examined. Including the high adoption rate of open-source tools, a growing knowledge base is coming out of U.S. colleges and universities (Babcock, 2000). The issues and barriers including MySQL security, prototyping shortcomings, security, and privacy were identified and noted as important challenges.

In this study, the investigator developed a prototype online collaborative medical application using open-source tools, except the server used in the study. Table 1 shows the relationship between research questions, assumptions, problem statements, goal statements, and the prototype's procedure for solving the problems. Table 2 shows recent Linux deployment by commercial companies and the exploding open-source presence in the computing landscape.

# Chapter 2

# Review of Literature

## Sources Consulted and the General Format for the Literature Review

The investigator conducted literature reviews of Web applications, the current stage of open-source acceptance, prototyping, project management paradigms, and successful online collaborative projects. The investigator identified articles, case studies, and scholarly publications of the Institute of Electrical and Electronics Engineers (IEEE) and the Association for Computing Machinery (ACM). The literature review was conducted through searches of scholarly databases such as the ACM's digital library and the Internet.

The Internet is a collection of sprawling networks that link millions of computers to millions of people around the world. "This 'network of networks' allows those with proper access the ability to log in to remote computers, send electronic mail, transfer files, access global bulletin boards, or search the Internet for software and information" (Hill & Misic, 1996, p. 12). For the Internet searches, search engines such as AltaVista, Lycos, and Snap were used. The Internet databases involved a computer search using a specific keyword by extracting the text of a journal article's title, abstract, and other related words in the record. As stated by Krathwohl (1993, p. 115), " reviewing all the abstracts over many years would be utterly impossible for a human being but is easy for a computer." Trade publications ranging from communication of the ACM, eWeek, and IEEE Computer magazine were used as sources in establishing findings in the research.

The key words used in the searches included: open-source, collaborative computing, Web applications, security, object- oriented design, design methodologies, three-tier development, XML, prototyping, evaluation, medical ethics, and online issues.

Leedy, Newby & Ertmer (1997) explained the following:

> Those who do research belong to a community of scholars, each of whom has journeyed into the unknown to bring back a fact, a truth, a point of light. What investigators have recorded of their finding will make it easier for others to explore the unknown. (p. 71)

The investigator organized the literature review section into six main categories: (1) open-source model, available tools, and the Web; (2) online collaboration and the open-source revolution; (3) history of Internet security; (4) the Internet and medical data; (5) object-oriented design and prototyping; and (6) design and the implementation of the three-tier medical online prototype.

The following topics were reviewed in an attempt to provide a foundation and background for the research questions that are outlined in Table 1 in Chapter 1. The questions that are stated in each category support the research questions that will be answered later in this study:

1. Open-source model, available tools, and the Web
    - What software benefits does the open-source model offer?
    - What is the future of the open-source model and tools?
    - To what extent can the open-source revolution help in the development of secured applications?

2. Online collaboration and the open-source revolution
    - What are the benefits of open-source collaboration in relation to Web applications?
    - What are the benefits of the open-source revolution in relation to current computing environments?
    - What collaborative dynamics helped projects like Linux, Apache, PHP, and MySQL in their success?

3. History of Internet security
    - What is the nature of security on the Web?

- Can open source tools secure applications on the Internet?

4. The Internet and medical data

   - What are the issues affecting medical data online?
   - What components are necessary to support medical data online?

5. Object-oriented design and prototyping

   - What are advantages of the object-oriented model?
   - What are the issues with the object-oriented model?
   - What are the benefits of prototyping?

6. Design and the implementation of the three-tier medical online prototype.

   - What are the historical contexts of the three-tier model?
   - What are the advantages of the three-tier paradigm?

## Open-Source Model, Available Tools, and the Web

This section contains a review of the benefits of the open-source model and its use as a collaborative tool. The future of the open-source model and its effect on reducing technology cost and improving security and system reliability are considered. In the review is an examination of how the open-source revolution has helped in the development of secured applications.

### *What software benefit does the open-source model offer?*

The open-source model supports collaboration. Open-source collaboration ranges from a few individuals to worldwide efforts involving thousands of collaborators. The open-source model is a bill of rights for computer users. It allows users to make copies of an application, improve the application's source code, and distribute those copies.

These rights are important to software contributors, as they keep all the contributors at the same level (Perens, 2002). According to Perens, "Research laboratories have adopted the open-source model to share information critical to scientific investigation. Businesses are adopting the model because it allows collaboration with other companies to solve problems without the threat of anti-trust lawsuits. Others have adopted the open-source model to curtail another Microsoft dominating the computing landscape" (pp.1-14).

The Internet provides the open-source model its impetus for global collaborations that enables disadvantaged groups the chance to communicate directly with the world. The model facilitates participation by creating and supporting information technologies and systems that will touch the future of participants. The open-source community's general theme leans toward the rejection of hierarchies, mistrust of authority, and promotion of democracy and decentralization (Dibona, Ockman & Stone, 1999). There is also the issue of companies that exist solely to support open-source applications. Hollandar (2000) wondered about the new open-source business model. He asserted that many companies depend on installing and supporting open-source applications as their core business. The author stated that, " Some people wonder what will happen to the companies whose foundation is supported by open-source, when software improvements solve the ease-of-use need for an operating system such as Linux?" (p. 5). The new open-source business model allows third parties to sell installation services, packaged documentation, and variable customer support schemes that businesses and companies can purchase at relatively low prices. The companies are not software development companies; however, they specialize in open- source customizations for companies facing technical challenges of downloading and installing open-source software resources (Hollandar, 2000).

*What is the future of the open-source model?*

Due to the benefits that open-source tools give IT consumers in cost, security, and stability, more companies are adopting the open-source model (Fraternalli, 1999; Leavitt, 2001; Neumann, 1999; O'Reilly, 1999; Shankland, 1999), and yet in government organizations the majority of projects use the traditional closed, proprietary model. In tight economic times, open-source is a tremendous help to companies that want to reduce their IT budgets by cutting back on pricey and unreliable server operating systems from commercial vendors (Perlow, 2001; Railsback, 2001; Sonnenreich & Yates, 2000). Although critics complain that Linux lacks sufficient training material (Perlow, 2001), Linux commercial adoption is overcoming the educational deficiency in supporting its growing user base. IBM had committed to produce every one of their servers, PCs, and software products compatible with Linux. The rest of the industry is only now scrambling to catch up with IBM's more than $1 billion dollar investment (Perlow, 2001). Web-based applications are a driving force for open-source acceptance.

Web-based application implements business logic and its use changes the state of the business. The important focus of the Web model focuses on the business logic and state. Additionally, presentation issues are important and tend to be more artistic, and less concerned with the implementation of business rules. Web applications, like other software-intensive systems are typically represented by a set of models: A use case model, an implementation model, a deployment model, and a security model (Conallen, 1999).

The Web had been constrained by performance issues. Delivery time between the servers and browsers is as important as the resources required for delivering the data. Research on improving server performance online has focused on networks and protocols (Zari, Saiedian, & Naeem, 2001). Web servers behave differently under different loads.

Zari et al. (2001) suggested enhancements to application implementation and operating systems running on the servers. Network studies are normally focused on improving network infrastructure and protocol configurations to improve online performance. Studies on network dynamics have resulted in several enhancements to HTTP, including data compression, persistent connections, and pipelining (Zari, et al., 2001).

### *To what extent can the open-source revolution help in the development of secured application?*

The role and interest in the open-source model is not new. It is a radical idea to many opponents, with the belief that open-source is more prone to failure than closed software. Open-source software like the Apache Web server and the Linux operating system have demonstrated higher levels of reliability and robustness than closed commercial software (McConnell, 1999; Sonnenreich & Yates, 2000). Open-source software offers an opportunity to prevent the risk associated with proprietary software (Neumann, 1999). Mockus, Fielding & Herbsleb (2000) stressed the virtue of the open-source software development model. According to the authors, " The open-source style of software development has the capacity to compete successfully, and perhaps, in many cases displace traditional commercial development methods" (p. 363). Many articles provide interesting examples of how open-source software has a greater future for adoption by businesses than proprietary software. Sonnenreich & Yates (2000) reported on cost benefits of open-source operating systems like Linux and OpenBSD for building firewalls against intrusion detection, providing high security to companies.

Fraternali (1999) indicated that open-source adoption by businesses was increasing. The research on the open-source phenomenon revealed that open-source software is a rising star in technology today (Dibona, et al., 1999; Raymond, 2000). The year 2000 witnessed corporations opening up to the open-source community. Many companies are in search of applications that run independently of hardware configurations and are cheaper to implement than custom programs developed from scratch (Correia & Finlay, 2001; Finley, 2000). Some enterprises jumped on the open-source bandwagon in order to save money. It is possible for a company to save millions of dollars using open-source applications. For example, the purchase price of Windows 2000 Servers for a large company may reach a million dollars compared to the purchase price of Linux servers at no cost (Hildebrand, 2001).

Fascinated by the rapid development and growing sophistication of the Linux operating system, Eric S. Raymond, self-appointed chief advocate for the open-source movement, began studying the open-source development model (Drummond, 2000). Raymond found that while the corporate, mainstream, closed-source method (the cathedral model) of coding large programs like operating systems is bound by Brook's Law; the open source development process (the bazaar model) actually reverses it. Brook's Law states that programming work performed increases with direct proportion to the number of programmers (N), but the complexity of a project increases by the square of the number of programmers ($N^2$). Therefore, it should follow that thousands of programmers working on a single project should become mired in a nightmare of human communication and version control. In "the Cathedral and the Bazaar", it is explained that the open source model (the bazaar) overcomes this problem through customary central version control, mutual respect, and an army of developers and bug testers (Dibona, et al., 1999; Raymond, 2001).

Most opponents of the open-source model, based on the investigator's view, are relying on disinformation that may have nothing to do with their experience with the open-source model, but instead on some incidental disposition. It is ultimately unheard of in an information economy for some extreme capitalist to produce marketable applications worth billions and distribute them free. Logically, the application may have some problems, or the developers are playing some sort of a game. To the investigator, the open-source community is the new age philanthropist, contributing to society in substance that is beyond money. As Moorhead & Griffin (1989) have noted on dispositional views, "For a number of reasons, a person may decide that he or she dislikes a particular entity or a product of greater utility to society. That 'sort of a' person would be expected to express consistently negative opinions of the particular product to maintain their sic consistent and predictable intentions" (pp. 83-84).

Not everyone is enthusiastic about the open-source model. Commercial and proprietary shops worry about the open-source model's effect on their profitability and market share. If current trends continue on open-source acceptance, more companies will be coming into the foray soon. Some people see the world one-way, while others have a different outlook altogether (Early, 1986). The open-source model as a new revolution in the computing landscape is motivating different sets of factors between proponents and opponents. To many in the business world, power and wealth seem to motivate people. However, in the open-source community, power and wealth may not be the overriding issue. Murray (1938) presented the concept of the "manifest needs theory." The theory assumes people have a set of needs that motivates their behavior. The "manifest needs theory" details that several categories of needs are important to most people and that any number of needs may be operating in varying degrees at the same time (Atkinson, 1964; Murray, 1938; Wahba & Bridwell, 1976). The "manifest needs theory" is similar to Maslow's theory (1940), Maslow argued that human beings are "wanting" animals and arrange their needs in a hierarchy of importance, with the most basic needs being at the bottom of the hierarchy. Maslow ranked the needs according to their importance. He placed “physiological needs” as the most important. Once needs are satisfied they cease to be important, and security needs emerge as the primary motivation. The needs escalate up the hierarchy until self-actualization needs become the prime motivator, given that a previously satisfied set of needs does not become deficient (Maslow 1943).

This investigator assumes that the motivation of the open-source community can be understood through Murray’s manifest theory. Several of the needs that Murray (1938) believed and that were further expressed by Atkinson (1964) to be the most powerful motivations in Murray's manifest theory were adopted by the investigator from the “Personality Research Form Manual,” published by Research Psychologists Press, Inc., as shown in Table 3. Murray did not arrange needs in accordance with importance, but stated that all manifest needs are learned needs. Human behavior is influenced by needs and motivations. Behavior reflects an underlying need. There are two kinds of needs. The primary need has biological factors such as food, water, and avoidance of pain. Secondary needs are more psychological in nature, encompassing Murray’s manifest of needs such as power and achievement.

The need for power is based on the need to influence and dominate others, and achievement is a need in overcoming obstacles. The affiliation need is associated with spending time with others; the aggression need involves trying to direct others and their behavior; the autonomy need goes into overcoming obstacles and sometimes becoming rebellious; the exhibition need involves being the center of attention and enjoying an audience; impulsivity is the need to readily show feelings and express wishes; nurturance involves the comfort and the need to take care of others; and understanding is the need of logical and intellectual curiosity (Murray, 1938).

Table 3: Murray's Manifest Theory (Moorhead & Griffin, 1989, p. 113).

| | | |
|---|---|---|
| 1 | Achievement | Aspires to accomplish difficult tasks; maintains high standards and is willing to work toward distant goals; responds positively to competition; willing to put forth effort to attain excellence. |
| 2 | Affiliation | Enjoys friends and people in general; accepts people readily; makes efforts to win friendships and maintain associations with people. |
| 3 | Aggression | Enjoys combat and arguments; easily annoyed, willing to hurt others; may seek revenge on people perceived to have harmed him or her. |
| 4 | Autonomy | Tries to break away from restraints, confinement, or restrictions of any kind; enjoys freedom from people, places, or obligations. |
| 5 | Exhibition | Wants to be the center of attention; engages in behavior that wins notice of others; may enjoy being dramatic or witty. |
| 6 | Impulsivity | Tends to act on the 'spur of the moment'; gives vent readily to feelings and wishes; speaks freely. |
| 7 | Nurturance | Gives sympathy and comfort; assists others whenever possible; offers a 'helping hand' to those in need. |
| 8 | Order | Wants to keep surrounding neat and organized; dislikes clutter, confusion, lack of organization. |
| 9 | Power | Attempts to control the environment and to influence or direct other people; expresses opinions freely; enjoys the role of leader. |
| 10 | Understanding | Wants to understand many areas of knowledge; values synthesis of ideas, verifiable generalization, logical thought. |

In reviewing Murray's manifest theory, the investigator identified achievement, autonomy, and understanding as the set of needs that motivates the majority in the open-source community. The open-source community's need for achievement inspires participants to accomplish difficult tasks with high standards and excellence. The investigator believes the need for autonomy influence most open-source participants to break away from society's restrictions such as licenses and copyrights. Lastly, the need for understanding enables participants to investigate and excel in many areas of knowledge. The investigator believes the community's high cohesiveness had led to many accomplishments including the Linux operating system and the Apache web server. Moorhead & Griffin (1989) made the following statement:

> Little research has been done to evaluate Murray's theory; however, Murray believed that each need has two components, direction and intensity. Direction refers to the object or person that is expected to satisfy the need, and intensity represents the importance of the need. Murray suggested that several categories of needs are important to most people and that any number of needs may be operating in varying degrees at the same time. (p. 112)

As stated by Thompson & Strickland (1984), "There is a natural tendency for people to draw upon their own personal values, background experiences, beliefs, philosophy, and ambitions when choosing among alternative strategies and shaping strategic plans" (p. 69). The open-source community is comparable to socially responsible companies.

The open-source community and socially responsible companies are concerned and committed to the power of private endeavors to foster positive social change, whether in investing or computing. Studies show that social responsibility and economic performance are positively related. Sturdivant & Ginter (1977) divided 67 firms into high, medium, and low levels of social performance and showed that, by different measures of profitability, the most socially responsible companies were also the most profitable (Steiner & Steiner, 1988).

The open source model is poised to affect the strategic directions of many industries. Frequent and dynamic technological advances are not new to the information sector. Many advances and methods have transformed information technologies and subsequently affected a number of industries. The global transformation of industrial economies into knowledge and information economies has changed the mode of management. The information-based economy has shifted management paradigms from production volume to time-based competition and management of knowledge. Over 70 percent of the gross domestic product of the United States of America is the output of information, making the information worker a central asset (Laudon & Laudon, 1996). Information technology growth and change occur faster than companies can adjust. The power of information technology innovations, including open-source, has grown faster than the ability of users to apply the technology in a timely manner. The rapid adoption of the open-source model can dramatically alter cost, capital requirements, and the experience curve of an organization.

## Online Collaboration and the Open-Source Revolution

Online collaboration and the open-source revolution focus on the significant contribution the open-source paradigm has had on the Web. The paradigm explores the global collaborative environment and the team spirit that open-source has helped develop. Open-source projects such as Linux, Apache, PHP, and MySQL are examined.

### *What are the benefits of open-source collaborations in relation to Web applications?*

A review of literature was conducted to provide insight about open-source collaborations on the Web. The sources included books, journals and other scholarly publications. Web-based tools are the wave of the future. Many companies have been moving towards the Internet to communicate and collaborate to meet project objectives. The Internet has fostered the growth of Web-based collaboration, allowing all members of a team to communicate, collaborate, share documents, and perform transactions using standard Web browsers. In a keynote speech in Toronto, Canada, Tim O'Reilly (2000) talked about the rich environment of the Internet and why open-source was central to that environment. He said, "Open-source is about collaboration, not just about software licensing." Friedlein (2001) stressed the areas that require special attention when managing Web-based projects, such as the need to obtain content from a client as soon as possible and the need to have a flexible, creative development team. People are cooperating with each other in conducting collaborative tasks based on shared inputs supplied by the participants, with privacy as the primary concern. Online application development and collaboration is similar to traditional collaborative efforts. Implementation only succeeds when the team is able to restructure itself, and not just overlay the new effort on an old structure (Markus, 1983).

Most traditional software development teams are organized in a hierarchy known as closed paradigm (Constantine, 1990). Friedlein (2001) wrote about how to manage Web collaboration with telling force. The author explained each step, from project clarification to review and evaluation. Friedlein's method presented helpful instruction on Web collaboration, starting with rigorous planning and ending with concerns such as team workflow, meeting planning, and Web content consideration. Hampel & Keil-Slawik (2001) stated that, "In its basic sense, distributed knowledge organization entails the cooperative generation, management, and maintenance of artifacts embodying knowledge. These artifacts include documents, graphics, notes, and comments that are linked to learners and forming the basis for methods in supporting learning processes using suitable environments and tools" (p.11).

The authors developed a collaborative tool named "sTeam," a learning laboratory for studying forms of distributed knowledge modeled on the basis of the open-source paradigm. The "sTeam" system took most of its conceptual and architectural ideas from cooperative-support environments such as Multi-user Object-Oriented Environments (MOOs) and Multi-User Domains (MUDs). MOOs and MUDs are basic architectures for bringing learners together in a virtual environment for independent interaction and communication (Curtis, 1992). Curtis defines MUDs as software systems that establish connections between users and allow access to common databases. Users move within a virtual world and interact with objects that are placed in virtual rooms. Rooms are connected and navigation is allowed between rooms and exits (Hampel & Keil-Slawik, 2001).

Hampel & Keil-Slawik (2001) indicated that, "Apart from the informal synchronous communication between users, the interaction between people and objects makes MUDs unique collaborative environments, as compared to other electronic media like chatting" (p. 12). Recent work and the growing importance of collaboration on the Internet make MUDs and MOOs key characteristics of a persistent virtual world that facilitates theme-oriented collaborations. According to Mynatt, Adler, Ito & O'Day (1997), "Interactions that may relate to domain of training and education, MUD is becoming socially acceptable, and is accepted by many as a serious application for supporting cooperation" (p. 13). These arguments taken as a whole suggest that collaboration on the Internet and the open-source paradigm will enjoy increasing acceptance.

Online discussions can differ from traditional ones in many ways. For example, collaborators in an electronic discussion can play an important role in contributing to a group's expertise and can also distribute responsibility for learning and remembering new ideas (Brown, Ash, & Rutherford, 1993). However, online collaboration lacks verbal or signal actions compared to the traditional classroom environment. For example, the MMCR collaboration that is facilitated through the "ClassLeader," a teaching and collaborative tool used at Nova Southeastern University, is designed to capitalize on certain social features of discourse that reduce social inhibitions, such as shyness and interest in listening to others.

There are many researchers in the area of electronic communities. Riel and Levin have studied structures needed to organize and support good interactions for online learning and teaching circles, as part of research conducted with AT&T (Riel & Levin, 1990). The authors showed their successes and failures in building networked communities and summarized the results of their networking experiences into five important areas: 1) group organization, 2) task structure, 3) response opportunities, 4) response obligations, and 5) evaluation. Online collaborative environments are limited with respect to interactivity of all participants. Interactivity is key to online environments. It provides participants with cooperative communal experiences. Interactivity is key to both conventional and distance learning. Student interest in the learning process is highly constrained when the level of interactivity is low (Anido-Rifon, Fernandez-Iglesias, llamas-Nistal, Caeiro-Rodriguez, Santos-Gago & Rodriguez-Estevez, 2001).

As in almost all interactions, feedback from users is necessary for understanding a collaborative environment. The investigator, with permission from Dr. Levin during the summer of 2000, used the ClassLeader as a complimentary tool for an object-oriented class. The course is a graduate course of three credit hours. The students were allowed an hour each on Saturdays and Sundays to review class lectures and ask questions, for a total of two extra hours a week. Once the class ended, the assessment was very positive, as students' grades rose remarkably compared with the previous semesters from the same class and same instructor. As with previous classes, the investigator was the instructor. The number of students with an 'A' grade rose from 12% to 35% with the complimentary ClassLeader tool and the extra two hours each week helped students improve their understanding of the class material. The non-compensatory time was a factor to the instructor, who had no incentive to complement his onset class time with online collaboration that helped students' understanding and grades. The university does not compensate instructors for extra time weekly using the Internet.

*What are the benefits of the open-source revolution in relation to current computing environments?*

Open-source tools are superior and robust. Open-source has been argued as the best methodology for building better software that increases economic assets to organizations (Neumann, 1999; O'Reilly, 1999). Even within the framework of a basic cost control for organizations, Mockus, Fielding & Herbsleb (2000) believed that the open-source style of software development competes successfully with traditional commercial development and in some cases may replace the traditional development paradigm.

Fielding et al. (1998) believed that humans as social animals have a natural propensity to collaborate with one another, and the Internet enhances that aspect of the social need. To organizations with an information infrastructure in place, open-source tools on the Web are positioned to replace client-oriented commercial tools based on low cost of deployment, easier administration, and distribution enabled by the Internet. Cohen (2001) showed the relation between the cost advantage of open-source tools and business success:

> Open-source software enabled Joseph McElroy, co-founder of a New York-based business that defies business stereotypes, to become successful. The businesses have fewer overheads and using freely available tools is able to underbid big names like Microsoft. Running a business is creative. Open-source makes business sense. Why depend on a proprietary company, putting your applications in that company's hands, when you need to protect the business that you have built? Open-source enables the company to present a price point that can attract small-business customers who need to see a profit. It is all about providing a food chain that everyone can digest. (p. 8-11)

Cohen (2001) believed that the company, Everydayoffice, had the ability to charge lower prices because it used open-source for applications, server platforms, and databases. In 2001, the company scored an alliance with ISI Dentsu of America, a technology services company with roots in Japan, to partner on developing the Japanese market for the company service that includes banks, trading companies, ISPs, and telecommunication businesses. This led to partnerships in Japan to develop co-branded Japanese language versions of its Web-based products and services. Cohen (2001) commented on open-source database offerings, asserting that open-source database software has made its presence known in the business sector:

> Database deployments are important topics for business planners struggling to move business intelligence and transactions to real-time. The top database systems vendor options have been IBM's DB2, Oracle, and Microsoft's SQL Server, running on proprietary software engines. Last year, 2000, a number of events broke the mold in what is an added competition from emerging open-source software engines. Emerging as the bright new kid on the block were PostgreSQL and MySQL database engines. These open-source products and others like them are expected to whittle away at market share. (p. 8-11)

Proponents of open-source have argued for the adoption of the open-source model by the mainstream computing community. Proponents believe the open-source model and its leveraging of the Internet represent an alternative economic model for producing robust software that will ultimately reshape the multi-billion dollar commercial software industry (Dempsey, Weiss, Jones & Greenburg, 2002). Companies that are against the open-source model, rely on marketing tactics of support and functionality to support their antagonism. The fear of support and functionality is not an issue, as many mainstream companies now offer support and enhanced functionality for open-source products. Firms including IBM and Sun Microsystems have made huge financial and support commitments to open-source products (Babcock, 2000; Cohen, 2001; DuBois, 2000; Leavitt, 2001; Shankland, 1999). As stated by Berlind (2002), "IBM, once a purveyor of closed and proprietary solutions, has recast itself as the flag bearer of all things open and interoperable. IBM's interoperability efforts and open source donations include Eclipse and code for UDDI and SOAP efforts." The use and development of open-source software range from ethical reasons against commercial monopolies by the open-source community to philanthropic tendencies of developers.

The advantages of the open-source model include the following: free availability of applications, there is no universal restriction on how the tools are used, other developers can improve and distribute the code, there are no fees for modifications or versions, and anyone can use the current code base to start projects. Online collaboration is one of the exciting ways the Internet may be used to add economic value to society. Information-based trade makes up a sizable and growing portion of world trade. Open-source (free) software bears pragmatic advantages for global processes and economic value to world communities. The open-source advantage is based on the technological empowerment that such key components and information assets afford knowledgeable users regardless of economic means.

Information technology and online resources are becoming the fundamental tools for all development issues in developing countries.

The most successful open-source projects have been in the development of critical systems. These critical systems usually have a mass-market appeal. Many secondary applications depend on these critical systems to operate. One such system is the Linux kernel, a critical component that other applications must have to function. Open-source tools do not create dependencies on multinational corporations for support; however, commercial support is available for open-source applications, with users not tied to a single supplier or vendor.

*What dynamics helped projects like Linux, Apache, HP, and MySQL in their successes?*

A search of literature was conducted to provide a significant insight into open-source projects and the dynamics that led to their successes. The sources included books, journals, and other scholarly publications. Collaborative projects have been the method used in most important projects, including Linux, Apache, and MySQL. A collaborative effort online is normally made toward specific goals that share common tools, enabling communication and coordination from geographically disparate locations. Many open-source projects require a concerted effort from various developers sharing common tools and using the web to facilitate the collaboration.

The most notable open-source project has been the Linux operating system. Software source code is a form of intelligent knowledge that is analogous to other scientific discoveries. In the scientific community, scientists publish to enable other scientists to build on their results. Publishing one's source code in order to foster continued innovation in computing makes an open-source community contribution. The

Linux project started with Linus Torvalds, a Finnish student, in 1991. The project continues today as a coordinated team of core global volunteers with Linus Torvalds as the central coordinator and final decision maker on architectural issues (Dempsey et al., 2002). Dempsey et al. described how the Linux project leveraged the power of the Internet:

> As with other open-source projects, the project leverages the power of Internet communications to bring together a large number of developers in a coordinated effort. The credits file recently released the kernel list of about 190 names; however, some estimates placed the number of contributors at about 1200. Linux has been produced through a community-based development process that is opened to anyone with the right technical skills and is prepared to contribute to future development efforts. (p. 67-72)

The Apache Project started from webmasters sharing patches to the National Center for Supercomputing Applications (NCSA) web server, and developed into the most popular server connected to the Internet today (Behlendorf, 1999). The Apache Project is what Schach (1998) considers a build and fix model. According to Schach, "A majority of open-source projects begin their development life under the build and fix model because they are designed to fix a specific problem experienced by a programmer or systems administrator. If the product matures, it evolves into a full-blown product meeting user needs and satisfying design requirements" (p.3).

Rasmus Lerdorf created PHP as a simple program written in Perl that tracked visitors to his online resume. Later it was expanded to include access to databases. Over time people began requesting copies for their personal use and a collaborative open-source programming language was born. Other programmers, notably Zeev Suraski and Andi Gutmans, contributed code, and later a wider collaboration rewrote the PHP from the ground up (Medinets, 2000). PHP is a combination of a programming language and an application server.

Gilliam (2001) said the following about how many open-source projects stated:

> Most open-source projects start off with little or no funding. Most start as a solution to a problem by a programmer at his or her job. The Perl programming language was such a product that has matured significantly since Larry Wall first wrote the language in 1986 to generate web pages programmatically. (p. 4)

According to Medinets, PHP in 1999 had over 600,000 Web sites using the language. PHP has a thriving developer community, including big names like Mitsubishi Motors, Volvo, E*Trade, First USA Bank, the San Francisco Giants, and the San Diego Zoo. All the open-source projects, including Linux, Apache, and PHP have their individual unique practices; however, there are some commonalities. Having passed the 5.1 million Web site mark in December 2000, Brockmeier (2001) indicated that Web developers who are still unfamiliar with PHP must start now, as the installation base of PHP is growing. PHP is a rapidly growing Web technology that enables Web designers to build dynamic, interactive web applications, incorporating information from a host of databases, and including features such as e-mail integration and dynamically generated images. (Lea, Choi, Kent, Prasad & Ullman, 2000).

MySQL's roots began in 1979 as a database tool created by Michael Widenius for the Swedish company TcX, and evolved to an SQL client/server relational database system originating from Scandinavia. MySQL is not an open-source project because a license is necessary under certain conditions. In essence, MySQL is free, unless the user wants to make money by selling it or by selling services that require using the database. MySQL's popularity is not limited to the open-source community; it is portable and runs on most commercial operating systems and on hardware up to enterprise servers. Its performance rivals any database in the market today and can handle very large database records (DuBois, 2000).

The open-source communities are based on open collaboration, inclusiveness, and willingness to share one's knowledge. Many open-source projects are relevant to the thinking in the positive hacker culture. Raymond (1999) stated that it is clear that open-source puts the user in the driver's seat, dramatically lowering the cost of ownership and providing a great recipe for high reliability. For these good reasons, the author believes, it would probably not be long before buying closed-source software for an organization's key infrastructure may be considered the height of irresponsibility.

Most open-source collaboration is the quintessential example of distributed software development. Open-source development is available to a large pool of worldwide contributors and loosely controlled project teams. To cope with the demands of openness and the fluidity placed on the collaboration community, open-source projects have evolved their own methods and organization (Cubranic & Booth, 2001). Open-source projects like the Linux project are similar to structured open teams used in software development

## Internet Security Issues

The issues of managing online security and the anonymity of the Internet are reviewed. Preventable training and human factors contributing to Internet security breaches are identified and discussed. A review of recommended open-source security tools were identified to provide minimal protection to systems using unsecured channels like the Internet. Many factors go into making a computer system secure. The most basic factor is the physical side of the computer. The success of a security measure for a computer system depends on much care was used in implementing the chosen security policy (Cohen, 1995).

*What is the nature of security on the Web?*

Within the past few years, a lot of research has been done on the Internet and its associated security. With the advent of the Internet, the issues of managing security online became important. The anonymity of the Internet creates generous opportunities for misbehavior by users. The Internet lacks a social system to create and promote trust outside the open-source community.

According to Oppliger (1997), the Internet is much less collegial and trustworthy compared to a "secured" Intranet environment. It contains all the dangerous situations, nasty people, and risk that one can find in society as a whole. The Internet's openness turned out to be a double-edged sword. Today, virtually everyone on the Internet is vulnerable, and the Internet's security problems are the center of attention, generating much fear throughout the computer and telecommunications industry (Chapman & Zwicky, 1995; Kent, 1993; Morris, 1985; Spafford, 1989). With the Internet, security policies must define activities that, for one reason or another, have been determined inappropriate. A security policy must be analyzed to determine whether access control, information flow, intrusion detection, or other forms of authentication are enforceable. Also, at what cost must security compromises be made. Intrusion detection is a major issue on the Internet. Intrusion detection systems complement other approaches to information systems security by providing a mechanism to detect attacks that were not anticipated or covered by other security mechanisms (Escamilla, 1998; Vacca, 1998). Many investigators address the questions of enforcing security mechanisms. Information security is seen as a management problem (Cohen, 1995). Technical safeguards constitute only one of many aspects of security that must be addressed from an organizational perspective (Adams & Sasse, 1999). Information security has as much to do with checking the references of your employees as it does with checking safeguards like access control, passwords, and cryptography. Complex and expensive security configurations as a safeguard for access control, etc. are useless if users compromise the systems by using passwords like their surnames, children's names, etc. (Adams & Sasse, 1999; Reason, 1990; Schneier, 2000). In recent studies, investigators have come to realize the importance of the human factor in security (Vacca, 1998).

Organizations have largely failed to consider usability, and thus have made increased security demands on users that are unattainable (Adams & Sasse, 1999; Menkus, 1988). Corporate users receive little training and support for security, causing preventable security breaches. Many users, according to Adams & Sasse (1999), showed no appreciation of how crackers break into computer systems, and predictably crackers chose easy-to-break passwords. The Bell-LaPadulla (1973) security model that enforces confidentiality fails to address human factors as they relate to predictable passwords. The model's two basic rules forbid a process to read data that has a higher security clearance than itself, and does not write data at a security clearance lower than itself. Internet usage and access by employees is another major business concern, where easy-to-guess passwords pose many security concerns and impact business revenues (Siau, Lim, & Shen, 2001; Siau, Nah, & Teng, 2002). It is becoming increasingly clear that the Internet is a critical component of the 21st century business landscape (Simmers, 2002). As workplaces have evolved, the adhesive that binds organizations has been through electronic channels.

*Can open-source tools secure applications on the Internet?*

Information is arguably the most important asset of an enterprise, after human capital, to most companies. Many companies are becoming aware of the risks their information assets face, and are beginning to find ways to protect their essential data. According to Guar (1999), the price of protection is always less than the cost of lax security. The author suggested the following security tools to help make it harder for hackers. Table 4 summarizes the open-source tools. The table explains five security tools that could help to improve open-source security.

Table 4: Open-Source Security Tools to Reduce Online Vulnerability

| | Security tools | Description and download site |
|---|---|---|
| 1 | Tripwire | Tripwire is used to detect unauthorized changes at a file/directory level. It tracks changes to key system files by maintaining a database on specified files. The tool reports any inconsistencies between the database and the current attributes of a file. These include file deletions, modifications, and changes ton access permissions *ftp://coast.cs.purdue.edu/pub/tools/unix/Tripwire/)*. |
| 2 | SUDO | The Sudo (superuser do) utility lets the administrator delegate root authority to users without sharing the root password. Sudo gives authorized users access to a subset of commands, files, and hosts on the network *(ftp://ftp.courtesan.com/pub/sudo/)*. |
| 3 | Ssh | Secure shell is the tool for remote access to system resources by many system administrators today. ssh uses public-keys to establish communication over the Internet *(ftp://cs.hut.fi/)*. |
| 4 | TCP Wrapper | This utility provides for tighter access control when users try to access services on a server. The TCP Wrapper utility monitors and filters incoming network requests for network services under the inetd configuration (*ftp://ftp.cert.org/pub/tools/tcp_wrappers/*). |
| 5 | Swatch | The purpose of this program is to scan the system log files to report security-related events or other events of interest. Swatch can be configured to send alerts to system administrators. The program uses a resource file to scan for certain events and to generate alerts (*ftp://ftp.stanford.edu/general/security-tools/swatch/*). |

The Tripwire software assured the security and integrity of data on the investigator's development server. The software on a server can notify administrators and users if, when, and how files have changed. "Superuser do" (SUDO) allowed the investigator to assign limited privilege functions on the system to selected users. Secure shell (ssh) provided strong authentication and secure communications over insecure channels like the Internet.

The transmission control protocol's wrapper (TCP Wrapper) is used as a second line for tighter access control to system resources. Finally, the swatch program was used to alert the investigator of security-related activities within log files. There are thousands of open-source security tools available on the Internet for security. These tools are the minimum recommendation for securing a system online, and the investigator believes that one's particular environment determines security needs. Haviland (1974) stated, "Culture is a set of norms that, when acted upon by members of society, produces behavior that falls within a range of variance the members accept as proper" (p. 264). Effective computer security policies require knowledge of the underlying environment and some degree of careful thought. The increase in numbers of Internet crimes is because users are no longer constrained by physical distance.

## Internet and Medical Data

The Internet and security issues with medical data are the center of both legislative and legal battles involving patient's privacy. The medical industry has to find a better security method to meet legislative and privacy policies. To meet privacy requirements, the industry must control security to its computing environment.

*What are the issues affecting medical data online?*

The major challenges of the Internet and medical data are the practice and publishing of sensitive data that are in conflict with privacy concerns. The role of the Internet in privacy involves its architecture, including the Internet's size, rapid changes, lack of coherence, and the interlinked nature of its structure. Healthcare data are rife with incompatible medical standards and coding schemes that require careful translations (Berndt, Fisher, Hevner & Studnicki, 2001). The authors indicated that healthcare data come from many sources and are delivered in many forms. The industry is highly decentralized with largely autonomous data collections, making data quality a significant challenge. Johnson (1997) expressed the issue of trust on the Internet:

> One cannot trust information and the individuals providing the information online for a variety of reasons. One does not know the individual with whom one is communicating. Online communication itself is vulnerable to sabotage. People [working online] are more likely to behave in undesirable ways when they are anonymous, so you cannot be sure what is going on. (p.64)

The issue of trust is compounded when anonymity is thrown into the whole online equation. The Internet has evolved very quickly and dramatically. The growth of the Internet and the technological and social innovation the medical industry has experienced is considerable in advancing medicine. The practical universal availability of information on the Internet has added many dimensions to medical complexities.

The argument of Corrales (2001) seemed to lend certain sanctity to the concept of a United Nations type of association to counter the dark side of the Internet. Corrales believed that the Internet's dark sides include new forms of criminal enterprises that are based on fraud, impersonation, identity theft, violent virtual communities, and groups linked to terrorist leanings. The author believed that global cooperation was a beneficial security tool in controlling Internet crime:

> Internet-based crime committed remotely from foreign countries will make it necessary to incorporate into laws between countries a method to cooperate in legal investigations and law enforcement while preventing crime havens from establishing themselves in isolated islands as currently available for taxes. Barring such cooperation, the Internet could render useless a particular country's legislation. (p. 53)

Such cooperation is of relevance to medical data across borders as the Internet explodes. The focus in deploying reliable, secured, and robust systems is highly desirable in most areas of healthcare such as clinics.

*What components are necessary to support medical data online?*

The Internet is the best medium to access medical data that are decentralized and in many forms. Huston & Huston (2000) discussed the possibilities of using the Internet to access medical information. The authors introduced "MedWeaver" as a prototype of an integrated service that combines simplified access to up-to-date information with fast information delivery. Medical collaborations online have many indirect implications, reflected in the following statement by Huston & Huston (2000):

> Telemedicine and the problem of fair compensation with doctors at one end and HMO's on the other is a problem affecting medical collaboration online. It is about who gets paid what and how to determine what portion goes to whom. Doctors regard HMOs with suspicion, involving holding back a large portion of premiums extracted from patients and their employers. (p. 93)

There are further implications on modeling medical systems. Armoni (2000) discussed the design of hospital systems. The author points out that it is necessary to understand how the current system works before designing a healthcare system. He recommends the use of visual representation for communicating with all the user groups. De & Ferratt (1998) presented issues from the planning and implementation of an information system designed to share patient information among seven Dayton-area hospitals. The authors detailed the process and outcomes about the perceived value and limitations of the system. Notably, the physicians were not involved in the planning process, even though they were the primary end-users. The planning process went on, with medical records personnel claiming to speak for the physicians, so it was not a surprise that the system was used less often than expected (De & Ferratt, 1998). The use of patients' own medical records as a basis for selecting, linking, and filtering information that provided a summary of their medical profile and set links to general supporting information on the net (Bental & Cawsey, 2002). The authors found that specific kinds of patients such as cancer patients value information that includes details from their own medical records more than general information alone. The authors implied that it is important to keep patients aware of privacy issues in order to bring privacy practices into patient information online. The Internet has the potential to allow information about patients and individuals with particular medical conditions and must forge an acceptable commitment to maintaining their rights to privacy and the confidentiality of personal information. Confidentiality and security of patient's information online remains one of the most important medical issues, and is difficult to provide without advanced modeling of the problem domain (Kilman & Forslund, 1997). With the advent of telemedicine and its global spread, the standardization of electronic medical records is needed to comply with new health regulations (Huston, 2001; Kilman & Forslund, 1997). According to Huston, "there are technological and administrative tools available for the safeguarding of medical information and electronic patient records. Unless the tools are implemented, they are of no use" (p.94). The author went further in noting that the management of security measures from the upper echelons of medical organizations is imperative. As the reliance of security of information systems grows in the industry, the potential for financial loss and compromises of patient confidentiality also grow.

## Object-Oriented Design and Prototyping

The object-oriented development methodology is examined, and the benefits from using it are noted. The view of software development as a costly undertaking is contrasted with more economic-based object-oriented development methodologies. Prototyping as a means for rapidly developing an application is reviewed within the context of the object-oriented development paradigm.

### *What are the advantages of the object-oriented design model?*

There are great benefits to object-oriented systems development. Investigators have done objective quantification of the benefits of object-oriented design. Using survey research methodologies, Johnson (2000) presented a concise preliminary assessment of object-oriented design advantages. Johnson summarized the findings of a mail survey conducted with experienced object-oriented and non-object-oriented designers. The survey attempted to identify the beliefs of the developers by asking participants to rate feelings toward object-oriented advantages and disadvantages. The author acknowledged limitations of the survey methodology, but the general trend observed was highly positive. Both respondents (object-oriented and non-object-oriented designers) consistently acknowledged the general benefits of object-oriented methodologies and downplayed the disadvantages (Johnson, 2000).

Henderson-Sellers & Unhelka (2000) indicated that, to implement a complex system successfully, one must use a development methodology like the object-oriented model. However, there are issues with some aspects of the object-oriented model. Henderson-Sellers & Unhelka indicated that the Unified Modeling Language (UML) is a work in progress, since some aspects of the notation are still evolving and are subject to different interpretations. UML is a notation, not a software development methodology. Concepts and notations are the two aspects of a methodology, but UML lacks a process (Blaha, 2001). The author believed that the omission of a process in UML was intentional, as it is more difficult to standardize a process than a notation. Cantor (1998) discussed project management frameworks based on the use of object-oriented technology and tools for project management. Cantor showed the flow of the object-oriented software development model from analysis to implementation. He emphasized more of the model's flexibility that allows developers to work along with changing customer needs throughout the development process. With the historical view of software development as a costly undertaking, better methodologies are being sought to reduce cost that can be caused by schedule delays, dynamic technology environment, budget contractions, and a host of other issues.

*What are the issues with the object-oriented model?*

The object-oriented models have gained wide acceptance during the last ten to fifteen years. The advantages of the open-source model include continuous peer-review, increased security protection in open environments such as the Internet, and they are free for everyone to use. Adoption of the object-oriented paradigm in application development has been successful in some cases where software engineering understanding is taken for granted, instead of programming constructs. There are many models within the object modeling concepts, and some are specific to certain types of applications. Many organizations fail to connect business requirements with software development and acquisition.

Such a pervasive disconnect is especially troublesome for applications that are critical operations (Blaha, 2001). The software process model should always adapt to the needs of the organization. The adaptation process is known as the method engineering (Kumar & Welke, 1992). Goldberg & Rubin (1995) believed that one model is not good for everything, especially in large organizations where different projects need relevant models. In many instances, the use of the object model failed to yield benefits because of ill conceptions of the development process required to make the model functional (Booch, 1994; Coad & Yourdon, 1991). Phases defined in the object-oriented models that are used to design and implement software applications are the acceptable steps in reaching a milestone. However, many people with strong backgrounds in object-oriented programming languages have almost no background in software engineering concepts and methods (Berard, 1993).

### *What are the benefits of prototyping?*

Prototyping is a process of building a new application rapidly and economically for the end-user's evaluation. Working with end-users with a shadow system allows the developer to help define the requirements, while having the user assume the role of domain expert. According to Laudon & Laudon (1996), using prototyping iteratively helps promotes active system design changes. Those authors stated that

> prototyping anticipates that users will change their minds; these changes can be incorporated easily and at a lower cost during the early development phase.

> Instead of generating detailed specifications and sign-off documents, prototyping quickly generates a working model of the application. Requirements are determined dynamically as the prototype is constructed (p. 443).

Using the object-oriented design methodology, the investigator combined prototyping because of the perceived complexities of the application. Prototyping was especially valuable because of the many aspects of the online collaboration requirements and security of medical data accessed over the Internet. Large systems must be subdivided so that prototypes can be built on the parts (Laudon & Laudon, 1996). According to Alavi (1984), a large system's subdivision may not be possible without a thorough requirement analysis to see how the different parts affect each other. User needs and behavior are not entirely predictable and are strongly dependent on the context of the situation. A prototype enables users to react to the parts of the system with which they will be dealing (Gould & Lewis, 1985; Laudon & Laudon, 1996). What is right with prototyping is that it works in many applications and serves a prominent role in software development. What make prototypes successful are the people behind the prototype (Berghel, 1994). The success is dependent on the technical and management people involved in the development effort.

## Design and the Implementation of the Three-Tier Medical Online Prototype

The three-tier development architecture that splits the software into three distinct software entities is examined in this section. The benefits of the three-tier architecture used in the investigator's medical prototype are noted and compared to other tier-based developments. The three-tier architecture's widespread acceptance and the future of vendor neutral standards are reviewed.

*What are the historical contexts of the three-tier model?*

In the early 1980's, American National Standards Institute (ANSI), in collaboration with the University of Minnesota, defined the three-tier architecture for building portable systems. This architecture divided data processing into presentation, processing (functionality logic), and data. The architecture considered the role of each of these data processing layers, within the framework of two popular client/server architectures. The first is the two-tier architectures that assign the presentation layer with most of the non-database processing in a single client application. The two-tier is much easier to realize and maintain. All code is executed on the client; no additional programs need to run on any machine. Changes affect only the program running on the client (JBDC White Paper, 1997). The robustness and ease of use of the two-tier development tools dramatically decrease initial development time; computing organizations paid a penalty when trying to update functionality simultaneously. When trying to integrate systems or migrate from proprietary development tools, problems are encountered using the two-tier architecture, including maintenance cost (Deitel & Deitel, 1999).

Three-tier architectures split the three layers into three distinct software entities. The architecture requires more planning and support, but can reduce development and maintenance costs over the long term by leveraging code re-use and flexibility in product migration. Three-tier architectures are also the most vendor-neutral of the architectures considered and thus can facilitate the integration of heterogeneous systems. The rapid growth of the Internet and the cost effectiveness of collaboration it has brought to the computing environments are creating unprecedented drives to develop large-scale distributed applications (Joshi, Walid, Ghafoor, & Spafford, 2001). The three-tier architecture used in the investigator's development prototype attempts to overcome some of the limitations of the two-tier scheme by separating presentation, processing, and data into separate, distinct software entities (tiers).

The same types of tools can be used for presentation as are used in the two-tier environment; however, these tools are now dedicated to handling just the presentation. When the presentation client requires calculations or data access, a call is made to a middle tier functionality server. This tier can perform calculations or can make requests as a client to additional servers. The middle tier servers are mostly coded in a highly portable, non-proprietary language such as Java or PHP. The middle-tier functionality servers are multi-threaded and can be accessed by multiple clients, even those from separate applications.

*What are the advantages of the three-tier paradigm?*

For organizations servicing customers with rapidly changing environments, three-tier architectures can provide significant long-term advantages through increased responsiveness to business climate changes, code reuse, maintainability, and ease of migration to new computing platforms and development environments. The choice of architecture dramatically affects the development time, future flexibility, and maintenance of an application. The underlying infrastructure must be able to support the inherent complexity of the architectures' distributed processing that is seamless to the end-user. A development project that uses three-tier applications may deploy with greater speed than two-tier systems. The amount of the middle-tier code can be re-used from previous applications. The speed advantage becomes a factor in the three-tier architecture when a sizable portion of existing logics used. Maruyamaa, Tamura, & Uramoto (1999) agreed that the three-tier configuration is the most popular way to build new applications.

The novelty of the three-tier architecture is its ability to improve host-to intranet/internet performance and its ability to scale to thousands of simultaneous users. A three-tier architecture is a flexible way of organizing distributed client-server systems. In the simplistic approach suggested by the name, every client is connected to every server. In a three-tier architecture, an intermediate connecting layer is introduced.

Experience indicates that these savings can be significant, particularly in organizations that require separate but closely related applications for various business units. Re-use is also high for organizations with a strong enterprise data model because data-access code can be written once and re-used whenever similar access needs arise across multiple applications. Kean (1991) pointed out that a firm's long-term ability to compete is directly related to the reach and range provided by the firm's technical architecture. His suggestions for defining a platform include selecting architectures which (1) protect existing IT investments; (2) ensure the firm's ability to adopt new technologies; (3) provide integration of heterogeneous resources; and (4) accommodate emerging standards embraced by a broad base of firms.

The discussion of popular client/server architectures exposes the weaknesses in the overwhelming majority of current client/server systems. The advantages of two-tier systems may include adequate workgroup-level systems that can be developed rapidly and employ empowering interfaces. However, such systems lack the openness, flexibility, scalability, and integration provided by three-tier systems. The case for deploying three-tier systems will develop over time as tools mature and the momentum for vendor-neutral standards increases. There are a variety of research opportunities, including examining issues in migration from a two-tier to a three-tier system. The conceptual issues discussed in the research relate to the issues of development time, and studying how the level of complexity in three-tier systems acts as a barrier to its widespread acceptance.

## Summary

In this chapter, the research has shown shows how the distinctive competencies of online collaboration can arise out of the unique strengths of open-source products such as Linux, Apache, PHP, and MySQL. Research laboratories have used the open-source model to share critical scientific investigations (Perens, 2002). The chapter examined the benefits of the open-source model. The open-source model as a collaborative tool helps organizations reduce technological cost, improve security, and resolve system reliability issues. Open-source has been considered the best methodology for building better software that increases economic assets to organizations (Neumann, 1999). The chapter reviewed the Murray's manifest theory in relation to the motivation of the open-source community. The needs for achievement, autonomy, and understanding were identified as the primary motivators of the open-source community (Murray, 1938).

The anonymity of the Internet was reviewed and Internet security breaches identified. The Internet is not trustworthy and contains all the dangerous situations, people, and risk as in society as a whole (Oppliger, 1997). The chapter introduced the benefits involved object-oriented design methodology and prototyping as a means of developing complex applications. The object-oriented model allows developers to work along with changing customer needs throughout the development process (Cantor, 1998). Prototyping serves a prominent role in software development (Laudon & Laudon, 1996). The benefits of the three-tier architecture were also examined. Three-tier architectures split the three layers into three distinct software entities. The benefits include code re-use and flexibility in product migration. The three-tier is the most popular way to build new applications (Maruyamaa, et al., 1999).

The open-source community's success is due to the fact that it is practical. Most distributions of open-source products provide the community with safeguards to their intellectual properties, while an adoption boom by commercial powerhouses is taking shape. The Apache Software Foundation is the only open-source organization invited as a member of the powerful Java Community process, a decision making body for Java resources. It is a group that decides where the future of Java will be and what standards should be included in Java development environment. Apache Software Foundation is uniquely placed to speak for the open-source community. Its pragmatic approach to issues and its obvious technical excellence making it one the most important influences in the software development community (Rubinstein, 2002). According Rubinstein, "Apache has made significant inroads; the group claims its Webserver is 40 percent faster on Windows than Microsoft's own IIS. The reason companies are sitting up and listening is that Apache is the world's most popular proven Web server" (p.1). Based on the degree of influence Apache had achieved, Tim O'Reilly of O'Reilly & Associates stated that, "Apache [Software Foundation] told Sun Microsystems to treat open-source as an equal. It shows the moral force of a reasonable participant in a process" (p. 12). The part of what makes open-source Apache important to commercial vendors is the licensing structure. The liberal license allows companies to incorporate the work of the Apache into their proprietary products without returning their changes to Apache (Rubinstein, 2002).

No one knows exactly what the future holds, but there is little doubt that computing technology will be a large part of it. Young people will take their places in work environments that need independent thinkers who have skills in problem solving, analysis, communication, and teamwork. Some of them will be using technology directly as a basic part of their employment. Beyond the workplace, as citizens they will need to understand technology's products, interpret information, and choose from ideas that inundate their lives. Skilled use of technology is an important part of their future, but more important is the skillful use of their minds.

# Chapter 3

# Methodology

## Research Methods Employed

The focus of the investigator in this study was on developing an online medical management prototype to provide online collaboration, enable effective data communication, and allow access security that used open-source tools for implementation. The open-source development tools included PHP, MySQL, XML, Apache Web Server, and LINUX. The online medical prototype consisted of a three-tier application development prototype that provided a novel way of managing sensitive medical collaboration over the Internet.

The three-tier format is the most popular way to build dynamic applications (Maruyamaa, Tamura, & Uramoto, 1999) and database-driven Web sites (Feiler, 1999). The exact boundaries between the tiers are sometimes subjective. The three tiers separate the database from the application processing that allows reduced risk of incidental or deliberate corruption of the database engine (Blaha, 2001). The three-tier model is illustrated in Table 5. The table shows the architecture and services associated with the three-tier model.

The online medical prototype consisted of three components: a Web browser as tier one, an HTTP server as tier two, and a content (MySQL) database as tier three. Relational database management systems (RDBMS) are an efficient means of dealing with large dynamic data. RDBMS also provide essential characteristics such as robustness, integrity, consistency, and availability on mission critical data such as medical access over the Internet (Blaha, 2001).

Netscape communicator, a commercial browser of the Netscape Corporation, was used as the front-end browser. The design allowed all browsers to access the medical system. Medical providers and Health Maintenance Organizations (HMOs) can access the server and maintain medical information with the browser. The browser allows a universal interface between users and tier two.

Table 5: Online Medical Three-Tier Matrix

| **Tier** | **Architecture** | **Services** |
|---|---|---|
| Tier One | The graphical user interface (GUI); that is, the browsers on the user's machine. (i.e. Microsoft Explorer or Netscape Communicator or Application) | (1) Uses query expressed in XML/HTML as an HTTP request. (2) Receives XML or HTML from the tier two. |
| Tier Two | Applications program or programs that run on the Web server and processes that data (i.e. Apache, or Microsoft's IIS). | Analyzes HTTP request, converts to ODBC or JDBC query to the database. |
| Tier Three | The database engine (MySQL database management system) that stores the data that the second tier requires. | The database management system (DBMS) deals with the large data using SQL and metadata to extract information. |

The front-end server in tier two provided the primary logic for the online medical system. The server acts as the gateway between the users, the content database, and the security modules. The content database contained the actual medical data. The content database server was deployed on the open-source Linux operating system, with the popular open-source Web server Apache. The design of the system involved the development of a preliminary specification-driven application that was used as a prototype during the definition stage of development. A medical office and a medical insurance company serving end users reviewed the basic structural and operational requirements of the prototype. The investigator's prototyping approach encouraged productive involvement of end users, reducing the chances of later design flaws. The prototyping approach facilitated ease of modification and enhancement. According to Zucconi, Mack, & Williams (1990), the model-building process improves the developers' understanding of the system and helps identify requirements that may have been previously overlooked. The goal of using prototyping was to explore its experimental and iterative nature in setting requirements and rapidly reflecting them in the system. Success of a prototype depends on the people behind the prototype, including the user, the designer, and the programmer (Berghel, 1994).

## Design Procedures

To achieve the goal in this research, the investigator adopted the Object Modeling Techniques (OMT) to analyze, design, and build the Online Medical Collaboration System. The Object Modeling Techniques allowed the investigator to describe the Online Medical Collaboration System fully. The object modeling technique described the static structure of the system in terms of classes, generalization, and associations (Martin & Odell, 1992).

Table 6 shows the procedure diagram for the collaborative online medical prototype. Table 6 summarizes the features of the prototype. The matrix represents the relationship between the steps, the expected process, a success factor, and the description of each step of the process. The investigator followed the object-oriented software development life cycle. The life cycle starts with conceptualization and analysis as the early phases of development that is followed by design and implementation.

The software development life cycle had a sequence of well-defined stages as shown in Figure 1 (Blaha, 2001).

Table 6. Procedure Diagram for the Collaborative Online Medical Prototype

| **Steps** | **Object-Oriented Process** | **Success Factors** | **Description** |
|---|---|---|---|
| 1 | Concept | User involvement and management support | Define structure of physical model already existing in users' minds. |
| 2 | Analysis - the perspective of the real world. | Favorable attitude of users. | Recognize and use controls to open-source-tools such as XML and object design to map outcomes (input/output) and controls (process). |
| 3 | Design - how to build the application from the analysis | Choice of development methodology and scrapping undesirable /unfeasible solutions. | Revisit requirements and analyze. Two important requirements include the creation of graphical elements and their modification. |
| 4 | Walk-through - building user acceptance | Favorable attitude and achieved objectives. | Test the initial prototype with selected users group. Suggest additions and/or deletions |
| 5 | Implementation - actual database and code | Quality of management implementation. | Gear to common requirements and meeting users' needs and expectations. |
| 6 | Testing - suitability for actual use | All modifications made are user friendly as envisioned to meet prototype's requirements. | Assure characteristics such as high performance, efficient transaction processing and low maintenance cost. |

## 1. Conceptualization

*Actions in conceptualization*

Software development starts with users, management, or a system designer conceiving a need to replace a cumbersome business process with a better one. The concept arises from critical examination of business processes that impact productivity or that can be profitable and have a market. When an opportunity is identified for a new system, specifics and scope are determined (Blaha & Premerlani, 1998).

The investigator formulated tentative requirements by identifying the concise problem domain as the first step in the modeling process. The problem domain involved the identification of the users, sponsors, and the specific requirements for the analysis phase. The investigator determined that the online medical system, in the context of this study, was developed within the constraints of time, cost, and limited resources. The investigator developed specific statement of requirements for the online medical system as shown in Figure 1.

Figure 1. Problem statement of the conceptualization for the online medical system

Develop an online medical system to manage the collaboration and facilitate the activities associated with the care and treatment of patients. The system must store information about employers, HMOs, patients, and medical providers.

The following capabilities must be provided:

- Adequate access controls to prevent unauthorized access.
- Secured enough to be deployed over the Internet.
- Track procedures, diagnosis, visits, and treatments.

The investigator identified two important audiences, the financial sponsors and end users. The output of conceptualization is a statement of specific requirements that serve as grist for the subsequent phase of analysis (Blaha & Premerlani, 1998).

*Goal of conceptualization and background literature*

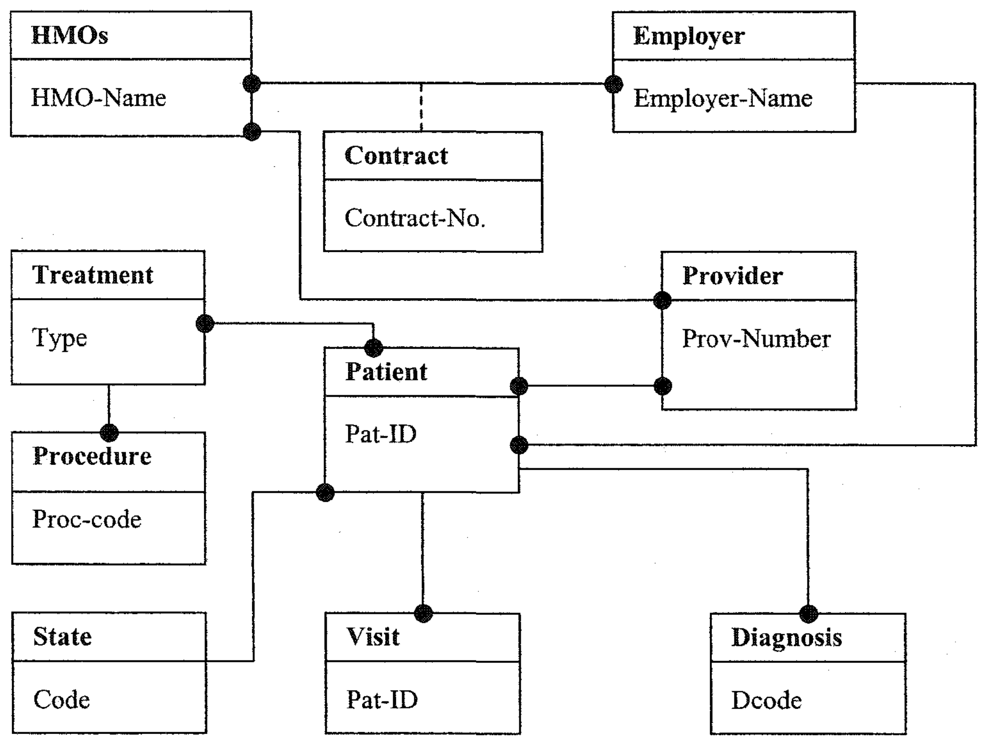
The genesis of the investigator's development of the prototype was recognizing that there was a lack of an effective collaborative environment for delivery of medical services. The first step was to develop an online delivery mechanism that medical providers, HMOs, and employers could collaborate in the delivery of better medical treatment to patients. The next step in the conceptualization process was to brainstorm. Brainstorming, according to Cayne & Lechner (1991), is a technique for eliciting ideas, decisions, or solutions to problems by concentrated, uninhibited discussion among a small group of knowledgeable persons. The next step was to generate ideas for a design of an online medical system. The investigator had a background in medical systems that helped in formulating a general idea on what was currently used in medical collaborations. Conceptualization is the first phase of the object modeling techniques (OMT) process of software development and deals with the essence of the application (Blaha & Premerlani, 1998).

**2. Analysis**

*Actions in analysis*

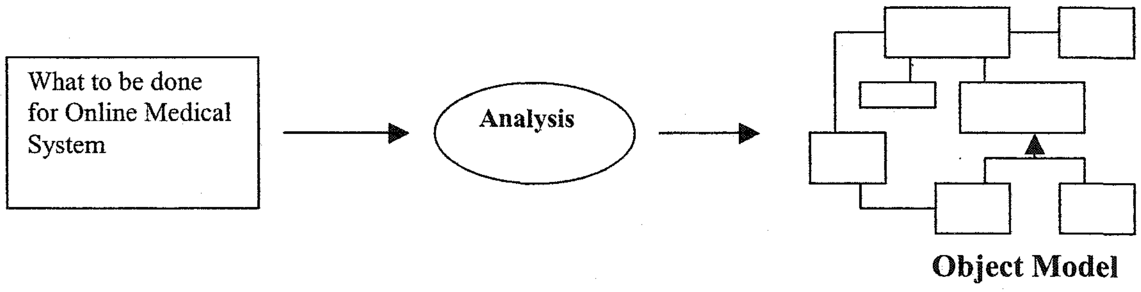
Analysis is to understand the problem requirements thoroughly and devise a model of the real world (Blaha & Premerlani, 1998). According to these authors, analysis expands on the requirements by specifying what needs to be done but not how it is done. The investigator used the conceptual understanding of the problem domain to construct models of the real world. All possible information sources were examined and domain experts in medical offices consulted in the effort to identify objects and related classes for the actual modeling. The investigator examined the static and dynamic aspects of the systems in terms of classes and objects. The conceptualization statement was decomposed into key object-oriented abstractions. The investigator generated tentative classes by extracting nouns and noun phrases in the problem statement. The sources explored in the analysis effort included the advice from problem experts, and artifacts from the investigator's prior medical systems experience. The abstraction involved the identification of possible classes and the subsequent elimination of redundant classes. After listing the classes, the investigator reviewed the dependencies between classes and determined their relationships. Figure 2 shows the graphical representation of the class diagram with the relationships between classes.

Figure 2. Class diagram of the online medical system



For example, *hospitals, providers, clinics*, and *dental offices* are synonyms; the investigator kept *provider*. *Clinics, hospitals,* and *dental offices* were redundant classes because the *provider* class was the most descriptive of all their characteristics. The class diagram (Figure 2) shows the interactions of the online medical system. HMOs may serve many employers, and an HMO may have multiple medical providers. Provider may give medical care to patients through diagnoses and procedures.

Figure 3. Analysis Phase of the Online Medical Prototype lists the original concept of the application and the models constructed to provide the basis for designing the application. Analysis specifies what is to be done but not how it should be done.



Developers begin analysis by modeling entities and relationships from various input sources (Blaha, 2001). Figure 3 shows the analysis phase that specified what must be done to achieve a functional prototype. During the analysis, available input was used to resolve ambiguities. Developers ask questions to refine and elaborate on the initial concept and devise a model of the real world (Blaha, 2001). The class diagram shown in Figure 2 was the investigator's static model of the online medical system. The concept was refined and elaborated to devise the model.

*Goal of analysis and background literature*

Modeling is the focus of analysis. Analysis prepares a model representing the real world that focusing on what must be done (Blaha, 2001). Blaha and Premerlani (1998) felt that the object-oriented strategy creates a powerful synergy throughout the development life cycle of a project by combining abstraction, encapsulation, and modularity. Muller (1997) advised that software implementation must take into account the organization, inter-relationships, and layout of structures in order to obtain the complex macroscopic behavior of the system being created. Consequently, building an application involves a series of 'divide and reunite' iteration; it is necessary to decompose in order to understand, and to compose in order to build (Muller, 1997). The analysis model deepens the understanding of the problem requirements and serves as the basis for design and implementation. The outputs of analysis were three models – the object, dynamic, and functional. The three models capture the essence of a system (Blaha and Premerlani, 1998). Blaha & Premerlani found the following about the models:

> The object model characterizes the static structure of things (objects). It looks at structure in terms of groups of analogous (classes), their similarities and differences. The object model is important for database applications because it concisely describes data structure and captures structural constraints. The dynamic model describes temporal interactions between objects. There is one state diagram for each class with significant dynamic behavior. The functional model defines the computations that the objects perform. (p. 7-10)

The models were refined and elaborated until they became coherent. A coherent model is able to enforce critical business rules and uncover inconsistencies (Blaha, 2001). Analysis specified what was been done not how it was done. Use cases were incorporated during analysis. Use cases have an intuitive appeal and may clarify subject matter expert's recollection (Jacobson, Christerson, Jonsson, & Overgaad, 1992). The design phase followed the analysis phase and dealt with specific applications issues.

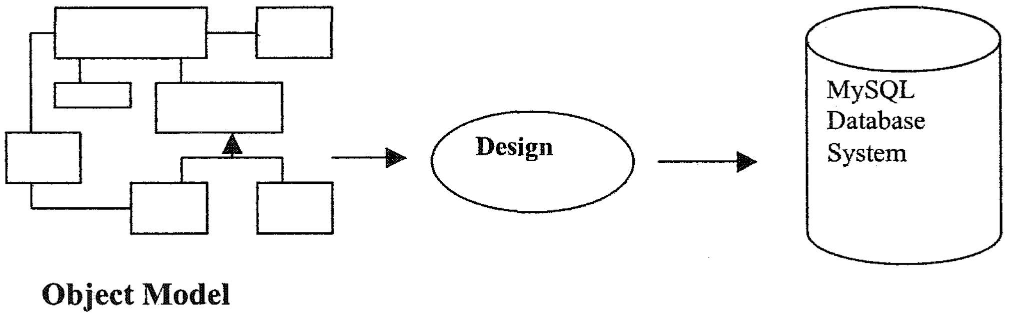
## 3. Design

### *Actions in Design*

Design allows the developers to devise a high-level strategy for solving the application problem. The investigator examined data management alternatives including in-memory data, files, and database management system (DBMS). The Investigator used DBMS for implementing the online medical system because there was much data to store and display but little computation. DBMS had been the appropriate choice for large and demanding applications (Blaha & Premerlani, 1998). The next step was to choose a specific DBMS paradigm. Relational DBMSs, object-relational DBMSs, and object-oriented DBMSs were the three specific models to choose (Blaha, 2001). The investigator's data management choice was the object-relational DBMSs. The open-source object relational DBMS (MySQL) offered many advantages. MySQL could handle objects and data intensive applications such as the online medical system.

The object model and system architecture, including the DBMS paradigm, were used to formulate the design phase of the medical online application. The next step in the design process was to choose an object identity approach. There were two choices, value-based identity or existence-based identity. In the value-based approach, primary keys are difficult to change. Another disadvantage of the approach was that many medical objects had no natural real world identifies (Blaha, 2001). The investigator's architectural choice of object identity was the existence-based identity approach. Existence-based identity was relatively easy to implement. Objects with no apparent attributes as primary keys were assigned an object identifier (Figure 10). The investigator added flow of identity notations to the object model, a characteristic of the value-based identity. The value-based identity had some advantages. Key attributes had intrinsic meaning to users, facilitating debugging and maintenance of the database (Blaha, 2001). In theory, primary keys are the sole record-addressing mechanism for a relational database and are normally defined for each table (Blaha & Premerlani, 1998). The design phase promoted and facilitated resources for the development of the system as shown in Figure 4.

Figure 4. Design phase presents the next step after analysis. The object model shifts the focus from the real world to the techniques of building the prototype.



*Goal of Design and background literature*

In the design phase, requirements were translated into logical and physical representations of the online application. It included handling both structured data (e.g., database records) and unstructured data (e.g., multimedia items) that support universal access by individuals with varied skills in the use of computers (Fraternali, 1999). Blaha and Premerlani (1998) encouraged a reconsideration of the model from analysis with an eye toward ease of implementation, ease of maintenance, and good performance during design. Design used transformation to simplify and optimize the object model. At the implementation phase, choice of data management approach was determined. The most widely used data management choices included organizing data into files, relational databases, and object-oriented databases.

## 4. **Walk-through**

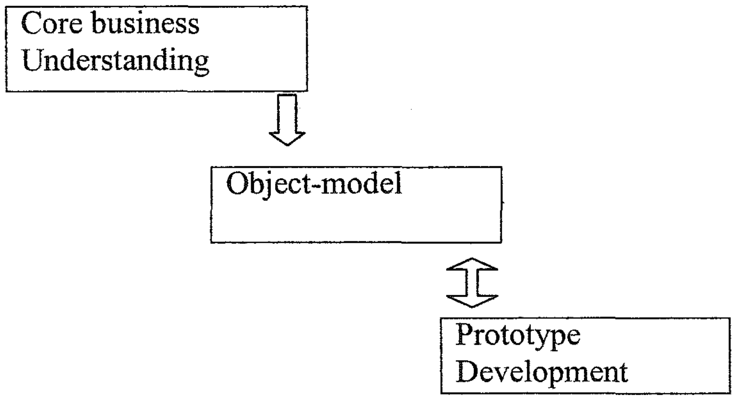
### *Actions in walk-through*

Walk-through was not an object-oriented requirement. The investigator added the walk-through step to build user acceptance of the application and to solve the prototyping aspect of the system. Prototyping allowed developers to repeatedly design software with end-users' input that expanded the scope of the project (Blaha, 2001). The investigator encouraged users to react to the parts of the system they were familiar with. The interactions of the domain experts with the system helped the investigator to correct development issues that provided target milestones. The investigator reviewed the details of the model from analysis and design, made adjustments to simplify implementation, and devised methods for computation and functionality. Figure 5 summarizes the purpose of modeling the prototype, adopted from Blaha (2001).

### *Goal of walk-through and background literature*

The business model in Figure 5, showing the interplay between the model and the prototyping flow, provided quality, coherence, and conceptual integrity for the online medical application. The prototype was broken into pieces using the object design and prototyping (Blaha, 2001), and the scripting language PHP was used to solve most of the database application tasks. King & Tester (1999) argued that creating environments of discovery was not a single persuasive strategy. Rather, when done well, these environments combine many strategies into a coherent experience.

Figure 5. Business intelligence from core business users and the object model incorporates what is learned.



The authors identified three components of the environment of discovery. The three components were: providing a fantasy environment for users, giving users control over much of the environment, and providing feedback to users for performing target activities. The investigator believed that tailoring information to users, as in the development of online collaboration, was a persuasive strategy. On the other hand, Tseng & Fogg (1999) stressed that credibility matters when computers provide data or knowledge to users. The authors elaborated that the popular culture from the past few decades in cinema and literature had cultivated trust in computing. Computers are often portrayed as infallible companions in the service of humanity.

## 5. Implementation

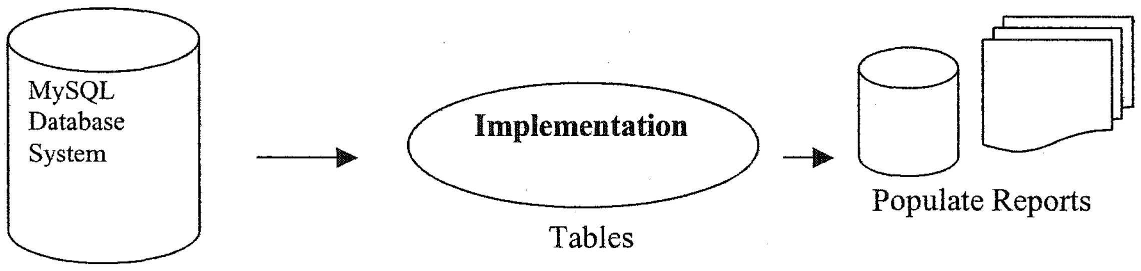
### *Actions in implementation*

At the implementation phase of the prototype, specific programming code using PHP were constructed by the investigator using prototyping. The implementation process involved the definition of primary keys for tables and domains. The investigator mapped the object model to relational tables and defined relational DBMS constraints that enforced the structure of the online medical model. With prototyping, one develops a portion of the system, uses it, and evaluates it (Blaha, 2001). After multiple iteration and evaluation by users, the prototype was considered complete. Figure 6 shows the prototype's implementation for the Online Medical adapted from Blaha (2001).

*Goal of implementation and background literature*

It is widely recognized that one of the current problems in the software industry is the communication gap between business end-users and software developers. This gap tends to limit software developers' knowledge of the business in terms of requirements, strategies, and operations.

Figure 6. The implementation phase of the online medical prototype deals with the actual database and programming code.



Implementation is the stage of actually putting the pieces together. It is the point in the development process where the designer must deal with the nuances of data management, programming constructs, and any other activity needed to construct the system (Blaha, 2001; Blaha & Premerlani, 1998). Testing was the next task after implementation. The system was tested before it was commissioned for actual use (Blaha, 2001).

## 6. Testing

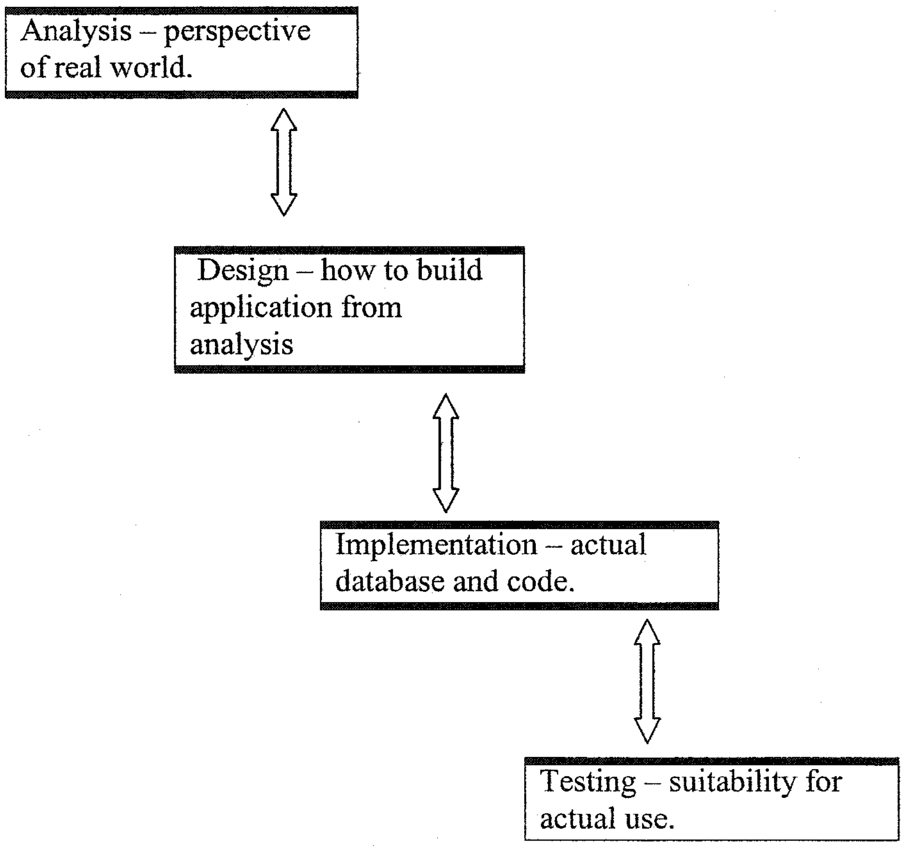
*Actions in testing*

After implementation, testing was undertaken before actual deployment for production. At the testing stage, the investigator ensured that the original project had been nurtured through the previous stages of the model, based on the defined design procedures (Blaha, 2001). The testing of the system allowed the users and the investigator to agree as to what extent the original business requirements were achieved. The investigator examined the object model and verified that the system delivered the required functionality. The application was tested on the implemented Linux platform and accessed over the Internet from medical offices to uncover possible accidental errors. The testing involved decisive factors: entry of initial data, testing access security, and access paths to patients' data. The application was considered complete after the testing. During the testing phase, the investigator tuned the database and created indexes to improve performance. The testing process is detailed in Figure 7.

*Goal of testing and background literature*

The object-oriented model should always adapt to the needs of the organization (Kumar & Welke, 1992). The investigator used the object modeling methodology to adapt to the needs of online medical collaborations. Phases defined in the object-oriented design methodology were the steps used in reaching and validating the online medical application's milestone and acceptability. In recent years, the Internet had been considered the ideal platform for developing applications that offered collaborations with users dispersed geographically. The paradigm is based on the Internet's powerful communication that allows multimedia capabilities, browsing, and open architectural standards to facilitate the integration of different types of content and systems (Myers, Hollan, Cruz, Bryson, Bulterman, Catarci, Citrin, Glinert, Gruden, & Ioannidis, 1996).

Figure 7. Iterative testing between development phases to achieve a working system.


Analysis – perspective of real world.
Design – how to build application from analysis
Implementation – actual database and code.
Testing – suitability for actual use.

Formal object-oriented design reviews were the core validation tool. During design and development planning using prototyping, validations of the application were identified.

The validation process followed the object-oriented design life cycle shown in Table 6. Technical validation was achieved through the continued interaction with business experts involved in the prototyping aspect of the application (Blaha, 2001). Testing uncovered accidental errors that had been introduced during the development phases (Blaha & Premerlani, 1998). Applications deployed over the Internet have conflicting security issues. Confidentiality and security of medical data remains the most important privacy issue (Armoni, 2000).

## Environmental Security Configurations

### *Action in environmental security configuration*

Access control from the Linux operating system, provided the online medical application the means to control individuals based on access privileges, day of the week, time of day, holiday, location, and privileges defined for each user group. Activating or deactivating individuals' access to the online medical system reduced intrusion compromises. The investigator developed a security policy to identify incoming traffic and authenticate the source as a trusted site using the originating Internet protocol (IP) address. Society is growing increasingly dependent upon large-scale, highly distributed systems that operate in unbounded network environments (Buchanan, 1996). Escamilla (1998) asserted that unbounded networks, such as the Internet, have no central administrative control and no unified security policy. The number and nature of the nodes connected to such networks cannot be fully known. Despite the best efforts of security practitioners, no amount of hardening could assure that a system that is connected to an unbounded network will be invulnerable to attack (Buchanan, 1996; Vacca, 1998).

*Goal of environmental security configuration and background literature*

The discipline of environmental configurations ensured that the online medical system delivered essential services and maintained essential properties such as integrity, confidentiality, and performance, despite the presence of intrusions. The investigator configured selected environmental files as the first line of defense for Internet traffic. The environmental control files used system level functions to determine whether external access requests could be permitted or denied.

The operating system security configuration involved the following environmental files:

I. /etc/hosts.allow – Access control lists was used for the purpose of controlling access. All incoming access requests were serviced based on the originating IP address status in the "host.allow" file (Mann & Mitchell, 2000).

II. /etc/lilo.conf - The boot loader for Linux had been set to read/write by root. A password is required to boot into a single user mode. (Nemeth, Snyder, Seebass & Hein, 1995). The objective was to restrict access to only authorized root users.

## Complimentary Security Tools for the Medical Prototype

*Action in complimentary security configuration*

Sensitive medical data must be protected from unauthorized access. Technical safeguards to a system must be considered from an organizational perspective (Adams & Sasse, 1999). It is becoming evident that the Internet is a critical business landscape as the information age evolves (Simmers, 2002). The Internet is the most efficient medium to access the decentralized nature of medical data (Huston & Huston, 2000). Complimentary security tools are necessary to protect medical data.

### *Goal of complimentary security configuration and background literature*

The prototype included the following open-source security tools that were loaded as separate modules to complement the Apache Web server:

I. Sudo - The tool assigned authorized users access to a subset of commands, files, and hosts on the network. A centrally configurable file controlled all configurations for the sudo command. The Sudo configuration file allowed sharing between different machines on a network (Mann & Mitchell, 2000).

II. /etc/ssh - Secure shell was the preferred tool to control remote access to system resources by system administrators. The /etc/ssh modules used public-key cryptography to establish a secure channel of communication over public networks such as the Internet (Schneier, 2000).

## **Validate the Prototype**

Allison (2001) suggested that the open-source development model had created software with significantly fewer exploitable holes than proprietary software. Allison believed that open-source developers' personal reputations were negatively affected in comparison to corporate reputations. The spin machines of corporate public relations are often available to companies to hide behind. Allison cited the "Love Letter" virus that exploited the security design blunder in Microsoft's email client, costing millions of dollars of lost productivity and lost data. The design blunder could have been avoided if the code had been open to peer review. Microsoft is still in business partly because of the public relations spin that repaired its image and economic motivation (Allison, 2001; Lipner & McGraw, 2001).

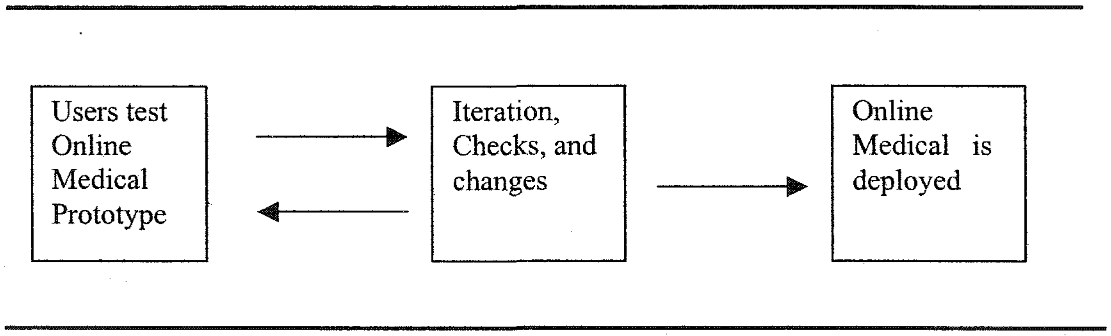
### *Actions in validating the prototype*

The goal of software validation in this study was to demonstrate that the completed product complied with established software and system requirements. An important task in the conceptual modeling process of information systems is the validation of the model. The investigator reviewed the programming code, user interface, and the databases structure against the intent of the object model and the online application using the business concept model in Figure 10. The validation task and objective was to check whether the model correctly and adequately expressed the requirements informally stated by users. Different techniques and tools had been developed to support validation task. They included the interactions between the object-oriented design phases and testing (Blaha, 2001). Because the research questions in this study were fairly broad in scope and general in their implications, the investigator tried to answer them using object-oriented model testing and prototypical feedback from the subject matter experts, the target users.

### *Goal of validating the prototype and background literature*

The investigator used prototyping to develop the application, use it, and evaluate it. The final prototype was delivered as the finished online application. Blaha (2001) concluded that the strength of rapid prototyping was that it provided frequent milestones and experimentation with troublesome development issues. In this study, a second benefit was greater maintainability as the model expressed the application's intent. Lastly, it helped communication by reducing misunderstanding and thereby promoting consensus among developers and users (Blaha, 2001; Blaha & Premerlani, 1998; Brooks, 1995). Figure 6 demonstrates through prototyping that all the specified functionality existed and that the application was made trustworthy. The ideas that inspired the original application had been nurtured through the phases. Original business requirements were verified and accidental errors corrected to achieve proper functionality (Blaha, 2001). The prototype was developed through a collection of freely available open-source development and programming tools from the Internet.

Figure 8. Completed system after testing and deployment of the online medical prototype.


Users test Online Medical Prototype
Iteration, Checks, and changes
Online Medical is deployed

Open-source tools are quite compelling for developers. The investigator believed that the use of open-source software achieved benefits in the areas of cost and quality. Table 7 identifies the tools and how they were used in the prototype. Linux was the operating system, PHP, the programming language, and MySQL, the database management system. One of the key components of the prototype was its user interface.

For many medical providers, the system was the only basis on which they could collaborate with their colleagues, insurance companies, and HMOs development process.

Table 7: Open-Source Tools to be used in the Prototype.

| Open-source tools | Description | Intended use in prototype |
|---|---|---|
| Linux 7.2 | Operating system bundled with security utilities. | Core operating environment for the Online Medical System |
| Personal Home Page (PHP) | Scripting language | The scripting/ programming language used to generate code and control three-tier implementation |
| MySQL | Powerful and secured database. | Database engine and Database Management System (DBMS) used to store and process data. |

To the investigator, the functional prototype was the first development for online medical collaboration using exclusively open-source tools. Because the project had limited resources and time, the prototype's evaluation was quick and inexpensive.

## Develop Evaluation Criteria

Boloix & Robillard (1995) indicated that evaluation approaches conducted after system implementations were used to evaluate the general value of the information system. The authors recommended the use of multiple-criteria evaluation approaches that included a subjective and objective evaluation matrix. A number of providers, HMOs, and other insurance companies could use the medical prototype that resulted from this study.

*Actions in developing evaluation criteria*

The evaluation criteria involved three candidate architectures for storing medical data. First choice was the storage of data locally at each provider's office; second choice was data storage at multiple regional sites; and third, centralized data storage as provided in the prototype. Local storage usage for all data storage and processing defeats the timely exchange of pertinent medical information. The regional clustering and single site collaboration used file servers to store and process data; personal computer clients at each point would provide the user interface and access data over telephone lines or the Internet. The investigator assigned the score of 50 for "want" and 100 for "must" to each decision criterion. The investigator treated the criteria as "wants" for the purpose of open evaluation to avoid spurious errors. Variables such as integrity, accuracy, and legal compliance of medical privacy are very important and will be regarded as "musts". The architectural choice with highest total score was the preferred choice. According to Bockle, Hellwagner, Lepold, Sandweg, Schallenberger & Thudt (1996), a computer system evaluation seeks to answer particular questions about the system, such as its performance or availability.

*Goal of evaluation criteria and background literature*

Blaha & Premerlani (1998) recommended that special care was required in deciding whether a criterion was a "must" or a "want." The architectural evaluation reflected the diversity of experience and perspective found in the user community and the healthcare industry in general. The investigator adopted an evaluation matrix developed by Blaha & Premerlani (1998) to determine data management architecture that included data choice.

Formulating system architectures is a matter of synthesis and must have clearly articulated goals (Blaha, 2001). The evaluation related the important factors of the system with storage requirements. The decisive factors used to determine a storage choice are summarized in Table 8. The investigator identified seven criteria for evaluating the online medical prototype. The first criterion was ease to use for the providers and HMOs, the major stakeholders of the system. The second criterion was data integrity and accuracy, conceivably the most important requirement in the medical industry. Virtually, all systems on the Internet are vulnerable to access and privacy security. Access security and patient privacy are seen as management problems (Cohen, 1995). Optimum security efforts were used to ensure that data was correct and up-to-date. The third criterion was extensibility, the possibility of additional uses of the system over time. The forth criterion was identified as the development effort that involved the use of open-source tools exclusively. It is the investigator's belief that development cost was not substantial because of the use of open-source tools. Open-source was the best methodology for building better applications that increases economic assets of companies (Neumann, 1999; Mockus, Fielding & Herbsleb, 2000). The fifth criterion was ease of maintenance and involved the requirements of regular updates and on going dynamic changes in the industry. With new Internet enabled technologies such as telemedicine, there is the need for standardization of electronic medical records to comply with global requirements (Huston, 2001; Kilman & Forslund, 1997). The sixth criterion was legal compliance. The system complied with privacy laws. The major legal aspects of medical data are the laws of privacy. Medical employers are aware of privacy issues to bring privacy practices to their employees (Bental & Cawsey, 2002). The last criterion involved the cost of development tools. Open-source tools are poised to replace client-oriented commercial tools because open-source tools are free, easier to administer, and available on the Internet for downloads (Cohen, 2001). The Internet is a warehouse of important applications and tools for business and education.

Table 8: Evaluation Factors for the Online System

| | Criterion | Requirement |
|---|---|---|
| 1 | Performance and ease of use | Responsive and ease for stakeholders acceptance. |
| 2 | Integrity and accuracy | Data must be trustful and consistent. |
| 3 | Extensibility | Must be able to adopt to new requirements |
| 4 | Development effort | Development cost must be reasonable |
| 5 | Ease of maintenance | Easy to maintain at a lower overhead |
| 6 | Legal compliance | Comply with privacy laws in the health industry |
| 7 | Cost of development tools | No development tools cost with free open-source tools. |

The evaluation also involved architectural choices of three storage alternatives. Table 9 shows the alternatives and related issues. The first was based on the ability to store data locally. The second alternative considered the ability of geographical storage. Geographical storage may be easier to maintain data-consistency of the master data than local storage. The last storage alterative considered was central storage, where a single copy is kept for updates.

Table 9: Storage Choices for the Online System

| | Architectural choice | Issues |
|---|---|---|
| 1 | Local storage | Requires minimum maintenance effort because there is no communication |
| 2 | Geographical storage | Easier to maintain data consistency. |
| 3 | Central storage | Cache data locally to improve performance and availability |

## Resource Requirements

The investigator purchased hardware and used exclusively open-source tools from operating systems to security and environmental configuration for the prototype. The latest versions of available open-source tools were used. Resources that helped the investigator achieve the functionality of the prototype were the availability and access to the following:

1. Academic and professional journals, textbooks, and papers on access security.
2. Professional group memberships such as IEEE, ACM, and others organizations relevant to information and integrated security technology.
3. Online and other Internet resources on access control and general security.

## Summary

In this chapter, the investigator presented the process that was used in building the online medical prototype. The online medical system manifested from inputs and outputs onto the various modeling phases into a finished application. Each phase of the object-oriented was examined and its significance as input to the next phase determined. The essential design procedures using the object-oriented methodology in the context of the online collaborative involved the sequence of conceptualization, analysis, design, walkthrough, and implementation. System testing was the next action, followed by security configurations to prevent unauthorized access and protect data integrity. The final stage was a validation of the application and identification of the evaluation criteria to evaluate the application for actual use. XML was used with HTML as the logical representation of data to different types of clients.

Ceponkus & Hoodbhoy (1999) explained that embedding XML within HTML could be useful because the XML document could easily travel with an associated script that knows how to make use of XML. The middle tier involved PHP scripts parsing XML through the Apache Web server. As an example, a request expressed in XML or HTML comes in as an HTTP request to the middle tier; it is analyzed, and is converted to a query or servlets. A request could be a patient entry, procedure update, eligibility inquiry, or a billing request. In conducting the study and producing the prototype using object oriented design methodology, the investigator drew on his years in computing and coursework at Nova Southeastern University. Regular supervision of the dissertation had taken place with the investigator's advisers and other faculty in the production of the prototype. The results of the research could serve to improve adoption of the open-source model by companies and educational institutions. In chapter 4, the results with the prototype are presented.

# Chapter 4

# Results

## Analysis

### *Overview*

The medical online collaboration prototype delivers a unique portal presentation that allows secure access to medical data. The advantages for the online collaborative system include the reduction of decision-making time and increased profitability to participating providers. The prototype, designed using the object-oriented design methodology, produced a data model that is both flexible and robust. The prototype provides secure access to data for authorized users. The discussion presented in chapters 1, 2, and 3 supports the cost and security benefits of the open-source model. The Internet and its architectural components have facilitated global collaboration in software development.

In Chapter 1, the investigator presented the scope and significance of the open-source model. That chapter explored software development issues relevant to open-source tools and software in the context of the Internet's collaborative environments. In Chapter 2, the review of the literature included the review of the open-source model, available tools, and the Web as it relates to online collaboration. Chapter 2 also contained the discussion of the history of Internet security, medical data, object-oriented design, prototyping, and the design and implementation of a three-tier medical prototype for online collaboration. The literature review highlighted the feasibility and technologies needed to develop the online medical prototype.

In the study, the researcher demonstrated the need for a secured collaborative computing environment for medical providers on the Internet. The constant onslaught of new and emerging innovations in information technology, such as open-source, has forced designers and managers to make difficult choices and then implement, deliver, and support these choices throughout business organizations (Rochard, Earl & Ross, 1996). The fast pace of information technology changes has created opportunities for research and commerce. Studies have shown that the information industry is challenging practitioners around the world to develop applications very rapidly (Boar, 1994; Carey, 1992; Mello, 1996; Paul, 1994).

Rapid information technology (IT) change can affect budgetary issues in many ways. These changes may include the need for new skills and continued training demands (Mello, 1996). Internal IT staff may resist new IT innovations and vendors may fail to supply expected support. Vendors may exaggerate capabilities or businesses may misunderstand marketing tactics and engender the unexpected need for more new information technology to solve selected problems (Benamati & Lederer, 2001). What can organizations do to avoid rapid technological changes like budgetary and staffing, while capitalizing on the benefits of the open-source paradigm? The use of open-source software benefits organizations through cost-effectiveness and robust application development solutions. The prototype in this study provides many benefits. It is a secure architecture that had achieved interoperation of services, and compliments the power of an open-source driven development.

The motivation of the researcher in this study, and the use of the object-oriented design, was to capture the viewpoints of all stakeholders. Chapter 3 documented the design process and components of the object-oriented methodology used in the development of the secured and cost effective prototype using open-source software and environment tools. In Chapter 4, the investigator presents the implementation results of the prototype. The chapter contains a discussion of the results of the completed investigation covered in the previous chapters. In Chapter 5, the investigator interprets and analyzes the results in Chapter 4. This discussion is organized into ten sections: analysis, security analysis, security customization, business concept model, prototypes architecture, online medical application, evaluation of the prototypes architecture, research questions, and cost of ownership.

## Security Analysis

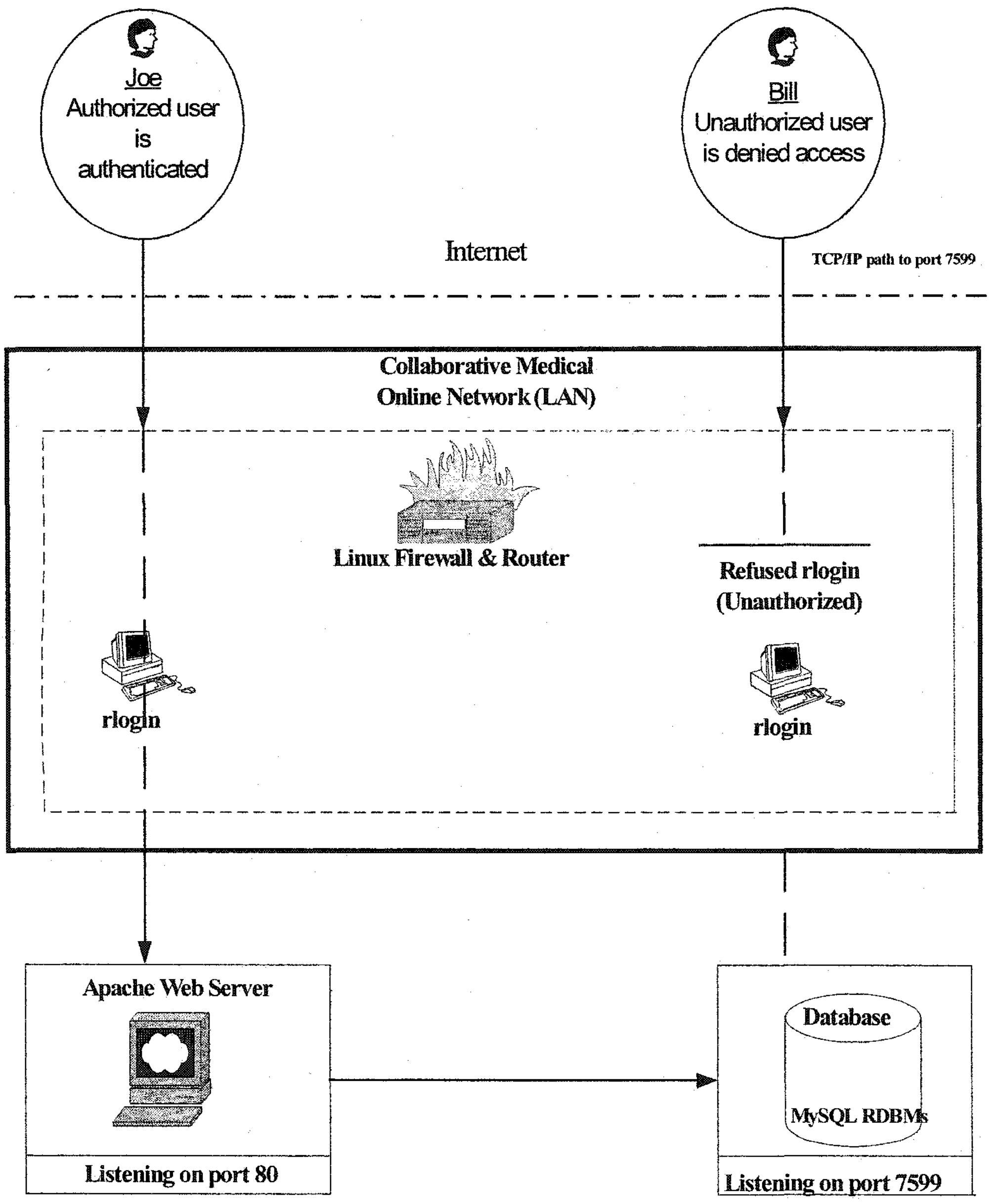
The purpose of security analysis is to protect a system from password cracker, vulnerabilities scanners, and Internet intruders. Analysis involves the design, configuration of security files, and testing of access paths such as the Internet and networks to verify and improve security. There is no single access control mechanism that provides the greatest overall benefit to users or a system.  Security customization is a means to restrict usage of known default access points. Unauthorized users typically use access points for entry. Having many access points increases the risk of authorized access to network services (Mann & Mitchell, 2000). Figure 9 shows the security configuration implemented to protect the online medical systems from unauthorized access. Figure 9 depicts the medical online systems with two scenarios of client's access. The authorized client (Joe) is authenticated and connects to the system. The hacker (Bill) is denied access to the system. The Apache server listens to port 80, the default web server port for connection request, and allows access only after the client is authorized and authenticated.

Standards methods have been developed to secure systems including open-source tools such as Tripwire, Sudo, and Secure shell (ssh). Tripwire is configured to secure the online application and maintain data integrity by detecting changes to sensitive files. Sudo is used to limit access privileges to system files. Secure shell protects the system from intruders by establishing secure channels for users. Security breaches on sensitive applications such medical can be disastrous in monetary terms (Huston & Huston, 2000). The Internet is an innovative communication channel that can also be used for destructive purposes. Rogue programmers have created malicious codes that have hurt individuals and businesses in monetary losses (Cohen, 1995). Stakeholders must decide what type of security is needed and to what level, it is needed. Each type of security configuration has its advantages and disadvantages. The degree to which protection is needed depends on the requirements for a particular system. To achieve a functional prototype, trade-offs had to be made between security and system performance.

### *Security customization*

There is no end to effective counter measures for addressing potential security breaches to medical data. Although effective system safeguards have been established to protect medical data, intrusion potentials are compounding with the rapid innovations in the computing environment. Medical information in transit over the Internet is vulnerable to interception. Several open-source mechanisms are available to enable administrators to control the dissemination of sensitive information. Secure shell uses end-to-end encryption. The Internet as a whole does not use secure links; so secure shell (ssh) must be used if encryption is desired across the Internet (Cooper, et al., 1995).

Figure 9. Security architecture configuration diagram for authorized and authenticated users.



Laws have been passed to insure privacy protections including the Health Insurance Portability and Accountability Act of 1996 (HIPAA).

The HIPAA act includes a provision that simplifies healthcare administration. The act includes a provision on how to maintain, protect, and transmit electronic data (Huston, 2001). According to Huston, the overwhelming majority of offenders of data security are employees or ex-employees. The most effective way to avert compromises by employees is to take effective measures, including dismissal. General security education on ethics and security responsibility can minimize unintentional breaches. A possible dismissal for intentional breaches is recommended to minimize employee security breaches (Huston, 2001).

### *Tripwire*

Tripwire is a tool that checks to see what has changed on a system. The program monitors key attributes of files that should not change, including binary signatures and increases in file size. Configuring Tripwire may involve balancing security, maintenance, and functionality. Tripwire, an open-source security tool allows the community to evaluate and add to the source code in order to tailor the tool to other platforms including Linux.

Linux is growing rapidly. Personal desktop users and Internet servers are a growing base for Linux. The growth of Linux adoption on the Internet exposes users to attacks. Linux users need the level of data and network integrity protection that Tripwire provides to identify changes to their programs, settings, and data. Tripwire's security logs benefits users through quick recovery from security breaches. Tripwire can manage and track changes to key system files by maintaining an information database for the specified files. Upon initialization, the Tripwire database contains a checksum, access permissions, and a creation date for the specified file set. Once the Tripwire database is initialized, the command itself can be run manually or at regular intervals via the UNIX "cron daemon." Cron is a Unix daemon that is concerned with the execution of tasks at a predetermined time period. Cron is run on the system every minute and tests whether the current moment matches any scheduled task requirement (Guar, 1999). The Tripwire tool reports any inconsistencies between the database and the current attributes of a file. These include file deletions, additions, modifications, and any changes in access permissions. It is important that the information database is configured and secured on a read-only medium to prevent unauthorized changes.

### *Sudo*

Sudo gives authorized users access to a subset of commands, files, and hosts on the network. A central file controls all configurations for the Sudo command. The idea is to share the Sudo configuration file between different machines on a network. The configuration file simply defines aliases for users, hosts, and commands, and denotes who has access to what commands, and on which host. The investigator configured Sudo to allow system administrator the ability to permit access to a small set of commands based on the role of the user. For example, a non-administrator can be authorized to add, delete, or modify users. Requiring a password for each administrative role, controls access to restricted system resources. The password itself may be cached by the system for a limited time for flexibility and ease of use. The time limitation and cached password could prevent abuse of the root shell. In addition, the execution of the Sudo command is logged. The default Sudo uses the /etc/sudo configuration file. The configuration file can be changed at setup and installation time.

*Secure shell (/etc/ssh)*

Secure shell is the preferred tool for remote access to system resources by many system administrators today. ssh uses public-key cryptography to establish a secure channel of communication over public networks such as the Internet. There is a server side and a client side to ssh. Once ssh is configured properly on the server, telnet and other remote-shell utilities run by the inetd daemon can be disabled. By default, ssh authenticates user passwords through their UNIX passwords database, pass phrases. The secure-shell configuration file (/etc/sshd_config) controls access to the system. The configuration file provides many options that can tighten up the access security on the system, including controlling root access, requiring a pass phrase for authentication, and turning off port forwarding on the server side.

In this study, access to the ssh server was controlled through different files, including the /etc/hosts.equiv, /etc/shosts.equiv, $HOME/.rhosts, and $HOME/.shosts files. For example, if a client machine logs in from a medical office and they are listed in any of these files on the ssh server, then the client is permitted access immediately without prompting for any form of authentication. The investigator's objective was to identify and authenticate requests for access. The strict security probes satisfies the study's security objectives and rules for accessing medical information. First, the accessing object must identify itself to the system. Second, it must prove that it the object it purports to be (Cooper, et al., 1995). The behavior is akin to that of many UNIX r* commands. However, any of the authorization modes could be turned off by directives in the /etc/sshd_config file, making the online system flexible. Additionally, the ssh server can verify the client's host key, as specified in $HOME/.ssh/known_hosts and /etc/ssh_known_hosts, to permit login for a few chosen clients. At the user level, a user can control access via $HOME/.ssh/authorized_keys. The file lists the public keys of only the known users. Furthermore, /etc/sshd_config (on the server) enforced authentication by the use of the public and private keys only. The keys required the ssh client to provide a pass phrase to the ssh server, and password-based authentication is disabled (Guar, 1999).

## Business Concept Model

Object-oriented modeling was the de facto choice for availability, reliability, and scalability of the prototype's design. The object-oriented model provides a robust and flexible service source to today's high-end database and Web servers. Security was considered core in the development. Increased deployment of applications on network servers makes security a priority (Gaur, 1999). According to Gaur, in the Internet age, open-source software operating systems such as Linux and FreeBSD had carved a niche in computing. The use of open-source security tools and software made systems secure. Guar cited open-source tools such as (1) Tripwire, a tool used to detect unauthorized changes to files and directories, (2) Sudo (superuser do), a utility that lets administrators delegate root authority to users without sharing the root password and (3) Secure shell (ssh), a preferred tool for remote access to system resources by many system administrators. The shell (ssh) uses public-key cryptography to establish a secure channel of communication over public networks such as the Internet.

Those who have accepted open-source tools are beginning to look seriously at their impact on entire organizations. Internet visionaries are talking about the impact XML will continue to have on Internet search engines, electronic commerce, intelligent agents, seamless roaming, file systems, electronic data interchange (EDI), push technologies, software distribution, data formatting, and more (Bradley, 1998). Figure 10 shows the business concept model used in the prototype. The following bullets defined the model for the online medical system:

- Employers' contracted with HMOs to provide medical services. The model shows that an HMO can have contracts with many employers and employers can contracts with many HMOs. The association between the HMOs and employers is a contract and employees.
- HMOs could have a group of employers with employees and medical providers could participate with multiple HMOs. Providers could have many patients and each patient must belong to only one employer at a time. Also, patients could have many providers for services such as medical, dental, and vision. Patients participate in treatments based on approved procedures.
- A patient may have visit records and associated diagnosis. A patient may occur only once in a state, such as the state of Michigan, at a time.

### *Transactions requirements*

The following transactions were undertaken to ensure that integrity of information was maintained to allow the day-to-day functioning of the online system. The functions had the responsibilities of the members of the collaboration team that included online medical staff, system administrators, providers, HMOs, and employers.

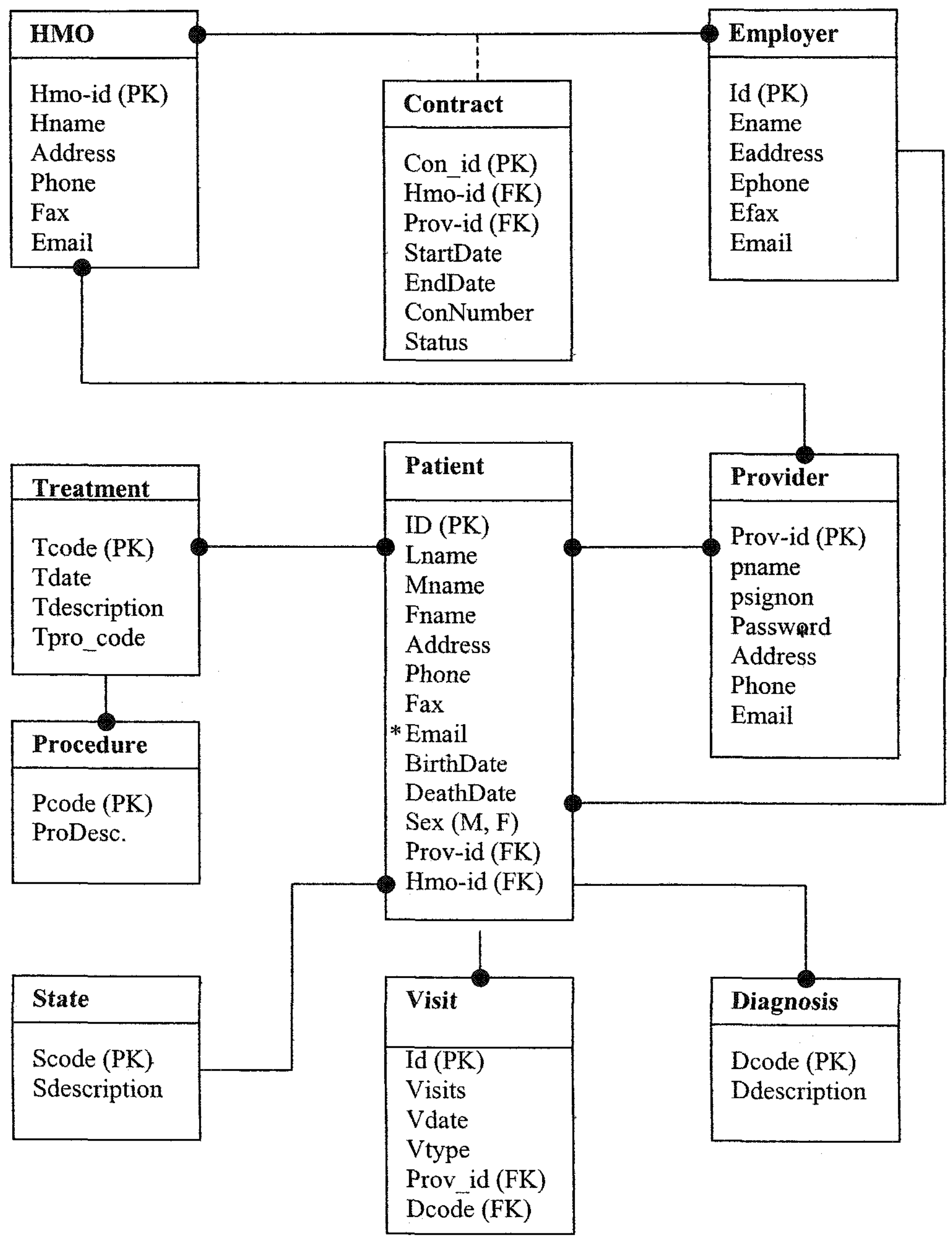
1. Create and maintain records and details of authorized users (System Administrators).
2. Create and maintain records and details of patients referred by an HMO (Online Medical Staff).
3. Create and maintain records and details of patients referred by employers (HMOs).
4. Create and maintain records and details of patients referred to a particular provider (Online Medical Staff).
5. Create and maintain records and details of patient's diagnosis and treatments (Participating Providers).
6. Create and maintain records and details of contracts by providers and HMOs (Providers and HMOs).

The business concept model captured the pictorial description of all the business concepts and their relationships in the online medical system. The model was the fundamental artifact that captured all the essential knowledge about the business at hand. It was detailed, concise, and forms the basic knowledge considered the backbone of the online medical application. The model helped in detailing the needs that had been communicated by business experts.

## Architecture of the Prototype

The prototype developed in this study consisted of a three-tier application development model used in system development. The model consisted of a Web browser as tier-one, HTTP server as tier-two, and a MySQL database and transaction system as tier-three. Relational database management systems (RDBMS) are efficient means of dealing with large dynamic data.

Figure 10. Business concept model of the online medical system.



RDBMS provide essential characteristics such as robustness, integrity, consistency, and availability on mission critical data such as medical information accessed over the Internet. Figure 10 shows the model for the medical application with the major objects that were mapped into relational database management system (RDBMS) constructs. A patient is a fact that depends on seven dimensions: Treatment, Procedure, State, Visits, Diagnosis, Providers, and Employers. An employer contracts with a health maintenance organization (HMO). An HMO has many providers and providers belong to multiple HMOs. A provider has multiple patients who receive treatments that have many procedures and diagnosis. A patient can visit a provider needed. HMOs have Hmo-id attribute as the primary key and Hname, Address, Phone, Fax, and Email and non-key attributes. An HMO can have many employers through a contract and employers can contract with many HMOs.

Netscape Communicator was used as the front-end browser. The design allowed all standard browsers the ability to access the medical system. The browser provides a universal interface between users and the middle tier. XML and HTML tags were used to represent logical data. One disadvantage of HTML is that, it was designed to represent only the presentation structure of documents. Rebuilt pages from a server involves a complete redelivery that places a higher burden on the server, because one is not getting only the updated information, but the entire data set (Maruyama, Tamura, & Uramoto, 1999). Maruyama et al. stated that XML solves all problems because an XML page is a logical representation of the data that can serve different types of clients. The logical representation is converted into an appropriate representation depending on the type of client.

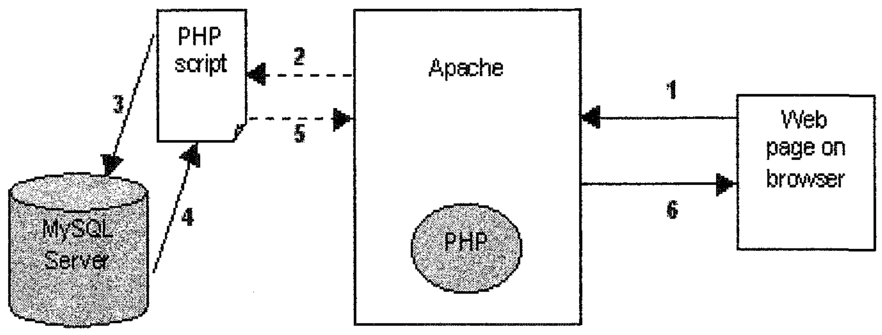
Ceponkus and Hoodbhoy explained that embedding XML within HTML could be useful because the XML document can travel easily with an associated script that knows how to make use of XML. The middle tier uses PHP scripts parsing XML on an Apache Web server. As an example, a request expressed in XML or HTML comes in as an HTTP request to the middle-tier; it is analyzed, and converted to a query or servlets. A request can be a patient entry, procedure update, eligibility inquiry, or a billing request (Reese and Reese, 1997). The query is sent to the database in the server tier where common Java API's are used to access database systems called Java Database Connectivity (JDBC). These developments included a Java Virtual Machine (JVM) installation. JDBC allows programmers the power of a high level interface (Chen, 1999). The open database connectivity (ODBC) allows access to any table format using the PHP script processes, where the results from the database are converted to XML or HTML.

## Online Medical Application

### *Overview of the process*

The framework for how the online medical collaboration system works and the interactions therein are shown in Figure 9. The framework involved the interactions of a request page on the browser that routes through the Apache Web Server that invokes a PHP script to manipulate the database and returned the results to the Web server onto the browser. Patient information was requested from a browser such as Netscape. Apache Web server received the client's request. The Web server invoked a PHP script that retrieved the requested information from the database engine (MySQL). The retrieved information was presented through the Apache server on the client's browser.

Figure 11. Components of the online medical system.



The interactions of providers and carriers was as follows:

1. Provider browsed the patient's information browser.
2. Provider added treatments, visits, and procedures
3. Carrier assigned weekly patients to providers
4. Provider's information was updated by Administrative Staff
5. Provider and Carrier contracts updated by Administrative Staff
6. Password and sign-on were managed by Administrative Staff

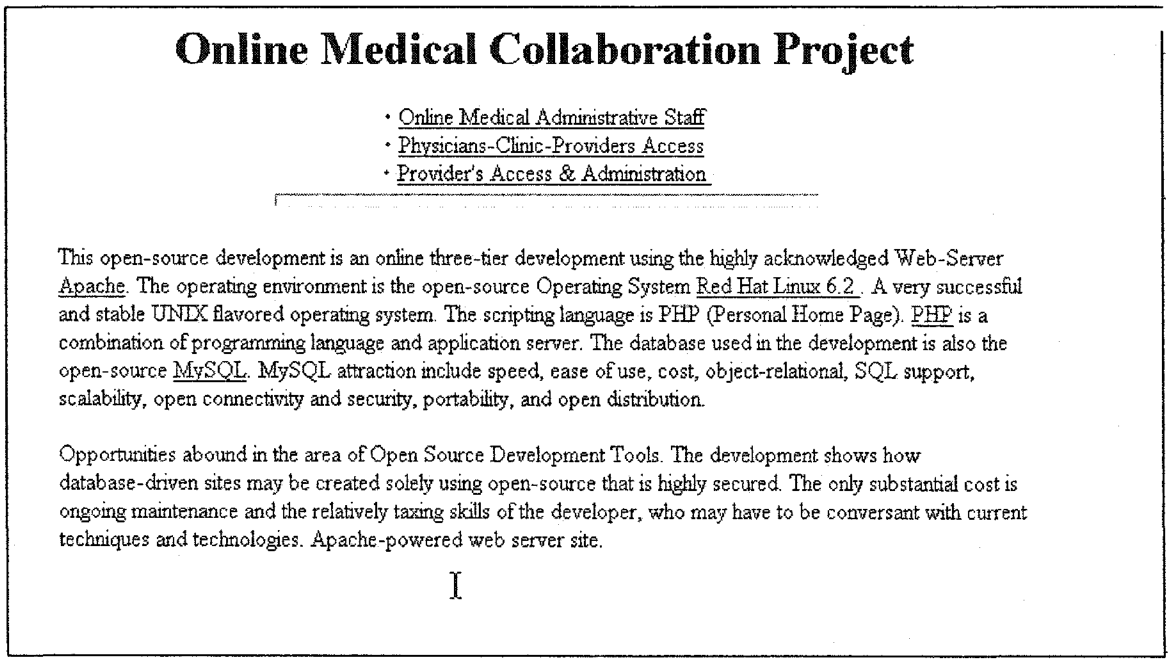
The Health Maintenance Organization (HMOs) assigned patients to medical clinics through the Administrative Staff menu. The patient lists for providers were updated. Patient processing involved the following:

1. Transaction went to carriers using Administrative System.
2. Carriers verified the provider and number of patients assigned.
3. Carrier paid a set amount for patients assigned to a provider.

*Administrative application*

Web directories were set up to require user-ids and passwords for access to web objects in them. The setup insured the Web server requested a user-id and password from the Web browser. The Web browser in turn requested the information in a pop-up window; after the information was provided, the browser remembers the user-id and password for the duration of the session and supplied them to the Web server each time protected objects were accessed. Figure 12 shows the index page for the online medical site. The menu contained three sections: online administrative function, provider's access for clinics and medical offices, and provider's administrative access.

Figure 12. Main menu of the medical online system.



**Online Medical Collaboration Project**

- Online Medical Administrative Staff
- Physicians-Clinic-Providers Access
- Provider's Access & Administration

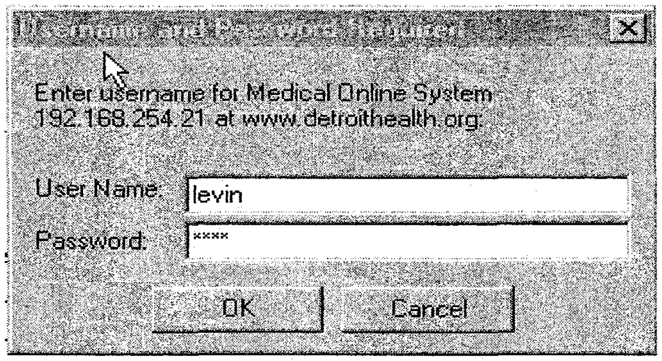
This open-source development is an online three-tier development using the highly acknowledged Web-Server Apache. The operating environment is the open-source Operating System Red Hat Linux 6.2 . A very successful and stable UNIX flavored operating system. The scripting language is PHP (Personal Home Page). PHP is a combination of programming language and application server. The database used in the development is also the open-source MySQL. MySQL attraction include speed, ease of use, cost, object-relational, SQL support, scalability, open connectivity and security, portability, and open distribution.

Opportunities abound in the area of Open Source Development Tools. The development shows how database-driven sites may be created solely using open-source that is highly secured. The only substantial cost is ongoing maintenance and the relatively taxing skills of the developer, who may have to be conversant with current techniques and technologies. Apache-powered web server site.

The function of the online medical administrative menu was to maintain, and administer the system. The online system administrators, and database administrators used the administrative subsystem. Clinics and medical offices used the provider's access for general clinical activities including treatment and diagnosis. The provider's administrative access was used to maintain access privileges and security.

The main menu was the center of activity for the online medical system. Each menu displayed a different view of the online application. Figure 13 shows the password authentication window for users that selected the online medical administrative staff. The profile allowed administrative users to access information regarding patients and treatment activities. The security structure contained information about users and access rights; its purpose was to allow a user to enter such information as name and password into the database. The provider could update or modify information related to patients provided by the contracting HMO.

Figure 13. Administrative staff menu.



[illegible]

Enter username for Medical Online System
192.168.254.21 at www.detroithealth.org:

User Name: levin
Password: ****

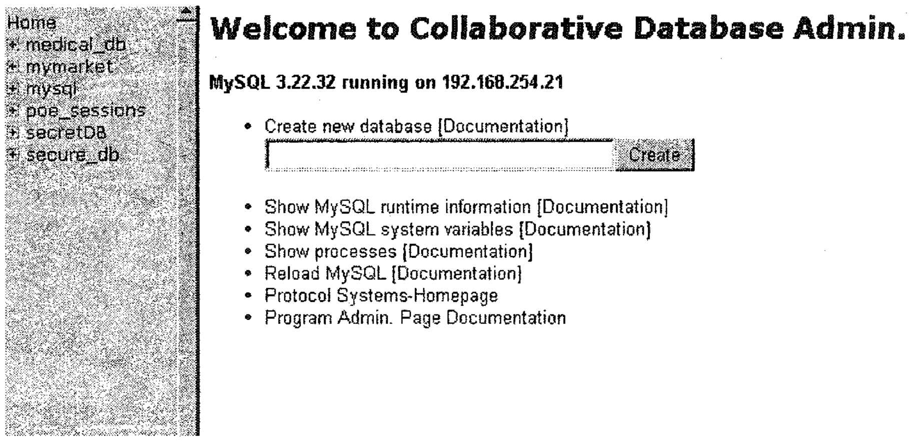
Upon authentication, the administrative user was shown a menu that summarized all the database tables the administrator was allowed to access. The administrator could create, read, update, and delete database tables. Accessing the administrator's profiles was limited to HMOs and system administrators. Internet traffic was defined and controlled in the Apache configuration file (httpd.conf) that allowed access from trusted sites. A trusted site was a client site that could be verified in the Unix-based operating file (/etc/hosts) for authenticating host names. In the medical industry, real-time processing involving patient information is key to data integrity. The operating environment of the online medical system was secure and reliable. The open-source Linux operating system provided a powerful configuration mechanism (firewalls) to control the type of Internet traffic that existed on a network. The capability restricted packets on ports based on specific screening as shown in Figure 9. The Linux operating system's enhanced security was used to argument the online medical security including user account expiration. The user account security included password rules that included upper and lower case letters, a non-alphanumeric character such as a dollar sign, and at least one numeric digit.

Figure 14 is a form used by the system administrators to access the administrative catalog. The administrative catalog was used to graphically manage and review changes at the database level. The administrative subsystem contained sub-forms for more detailed information reviews. Specified system level administrators from the HMOs used the access category to load and distribute patients to providers under contract. The category allowed for the following drill downs:

- Sub-tables and their contents
- Parent categories linked to the top category
- Content drilled downs for modification

The maintenance menu shows all the views of the database maintenance. The maintenance menus required administrative privileges to access. Figure 15 displays the functions used by the administrative staff to create, update, format, and enforce constraints. It also provided security and control mechanism for the online system.

Figure 14. Administrative menu after authentication.



Welcome to Collaborative Database Admin.

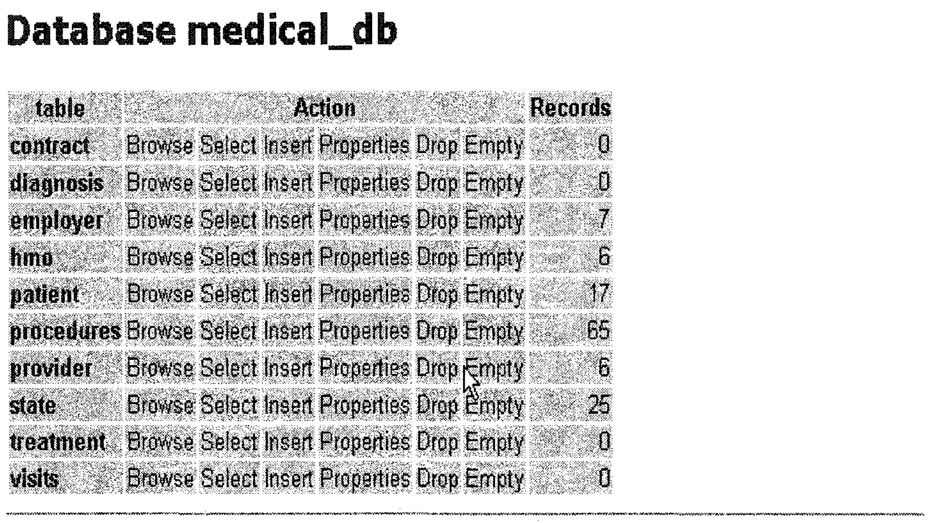
MySQL 3.22.32 running on 192.168.254.21

- Create new database [Documentation]

- Show MySQL runtime information [Documentation]
- Show MySQL system variables [Documentation]
- Show processes [Documentation]
- Reload MySQL [Documentation]
- Protocol Systems-Homepage
- Program Admin. Page Documentation

The ten tables directly corresponded to the ten class objects on Figure 10. The business concept model formed the basis of the ten objects transformed into relational constructs including tables and relational operations. Each table of the medical database retained its object name from the object model shown in Figure 10. The graphical menu defined actions that include browsing, selection, inserting, and deleting on the tables. Authorized users can view the number of records contained in each table. The administrator could restrict users to viewing data that pertained only to their predefined function.

Figure 15. Database maintenance screen.



## Database medical_db

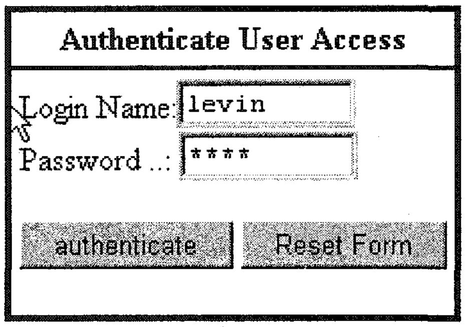
| table | Action | Records |
|---|---|---|
| contract | Browse Select Insert Properties Drop Empty | 0 |
| diagnosis | Browse Select Insert Properties Drop Empty | 0 |
| employer | Browse Select Insert Properties Drop Empty | 7 |
| hmo | Browse Select Insert Properties Drop Empty | 6 |
| patient | Browse Select Insert Properties Drop Empty | 17 |
| procedures | Browse Select Insert Properties Drop Empty | 65 |
| provider | Browse Select Insert Properties Drop Empty | 6 |
| state | Browse Select Insert Properties Drop Empty | 25 |
| treatment | Browse Select Insert Properties Drop Empty | 0 |
| visits | Browse Select Insert Properties Drop Empty | 0 |

Appendix A lists the medical online database tables and a descriptive listing of each table showing the name of the column, data type declared, and status. Combining two primary keys to create a multiple-field primary key were defined for classes with many-to-many relationships. Appendix A shows the details of the online medical database tables including table names, fields, data types, and related integrity constraints.

*Physicians, clinics, and providers access*

Figure 16 shows the general authentication menu clinical users could gain access to the maintenance screen. The maintenance screen allowed providers to update patients' information loaded from HMOs. Physicians were grouped by clinical roles based on specialty. A specialty could be in the area of urology, dermatology, and gynecology. Permissions were given to roles; typically, all users assigned to a given role got identical privileges. The online medical system used conventional business practice of separation of duties and responsibilities. The user profile was given a set of rules required for the given role. In addition, user-level security procedures were outlined and enforced.

Figure 16. General users logon authentication.


Authenticate User Access

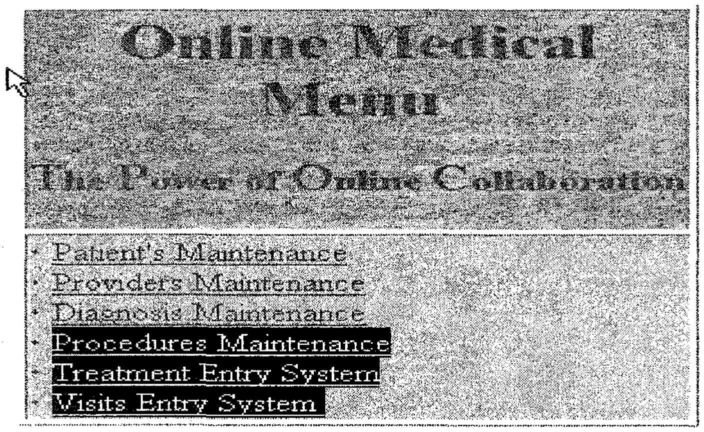
Login Name: levin

Password ..: ****

Physicians could look up archived patient information. The rules limited each physician to specific patients assigned to them. The system allowed emergency room physicians to retrieve a patient's medical history, allergies, and special instructions from a primary care physician. The rules for HMOs included the ability to review medical cases and check that diagnoses were reasonable. As an example, Jane, the office manager at provider A, was allowed total access to all patients assigned by insurer B to provider A. Jane had an access profile, and as with all office managers, her database views were limited to only the patients assigned to provider A.

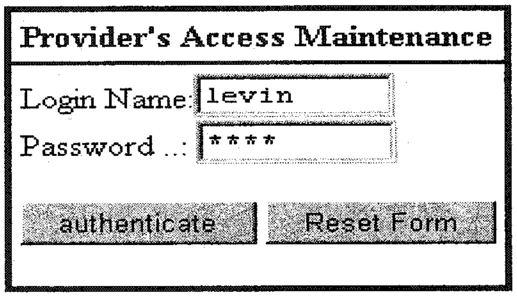
Security of access to menus was an important security need for the online medical system. Restricted menu access meant the online system met the requirements of a secured online system. Users such as office managers and physicians could update patients, providers, diagnoses, procedures, treatments, and visits as shown in Figure 17. The need to maintain confidentiality of passwords such as not sharing or posting passwords on computer monitors was emphasized. The investigator believed that behavior of users could be modified by training and constant reminders of individual responsibility in relation to the privacy of patients.

Figure 17. General users transaction menu.



To maintain effective access security, a second level authentication was implemented for advanced access and services. Figure 18 was used to control access to advanced security services. The application used centralized storage that allowed access from authorized and participating HMOs and providers. To maintain and enforce access security, log files were used to document all activities. The effort was to prevent and identify intruders masquerading as legitimate users. Logging was the security method for collecting information of controlled events such as logging into the system. Logging becomes useful when it could be used to analyze an incident after the fact (Cooper, et al., 1995). Because of security tracking, a second password dialog window was implemented in the prototype as shown in Figure 17 to assure security.

Figure 18. Advance security administration form.

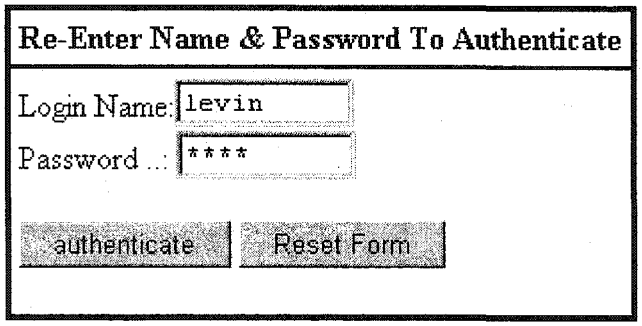

Provider's Access Maintenance

Login Name: levin

Password ..: ****

The major design decision concerned security validation that became important when access was granted to very important components of the application. Another decision was to make sure good data got into the system. Apart from several types of access validation, security to the maintenance menu was essential. Both the main selection and the sub-forms had record selectors because they were separate maintenance forms and contained core information for the collaborative system. A second form was displayed to authentication access to the security tables (see Figure 19).

Figure 19. The second authentication form.


Re-Enter Name & Password To Authenticate

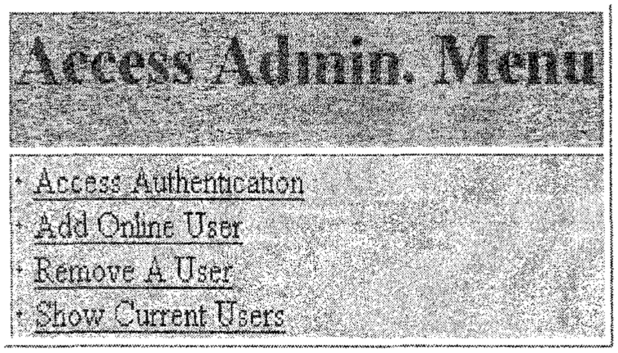
Login Name: levin

Password ..: ****

The second authentication helped restrict access by using the open-source filtering tool, TCP Wrapper. The TCP Wrapper process forced access control rules when connection request arrived. TCP Wrapper's actions were governed by rules specified in two environmental files, the /etc/hosts.allow and the /etc/hosts.deny files (Cooper, et al., 1995).

The password was recommended to be a code word for each provider's high-level access. Selected staff of a particular provider used the code word to access restricted system objects. After the second authentication, an access administration menu as shown in Figure 20 was presented. The menu gave the user the ability to change user profiles, add new users, delete user accounts, and print or view current users.

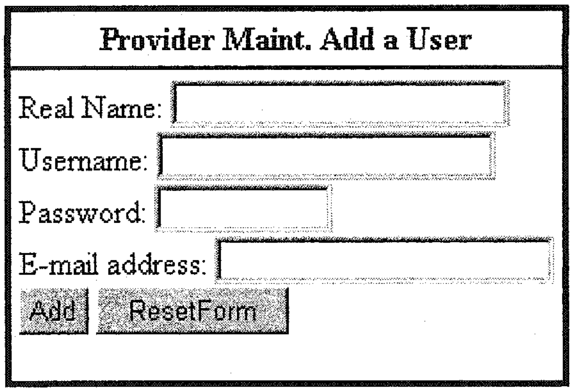
**Figure 20.** Security maintenance menu.



An authenticated user could view audit logs. Audit logs were used to track login activities (Cooper, et al., 1995). Figure 21 shows the add user screen. Add user maintenance allowed a provider to add new employees and make access changes as necessary. Figure

22 shows the user deletion screen. The form allowed providers to delete users from a provider's account. Provider A could not delete the employee of provider B.

Figure 21. Add user dialog screen.

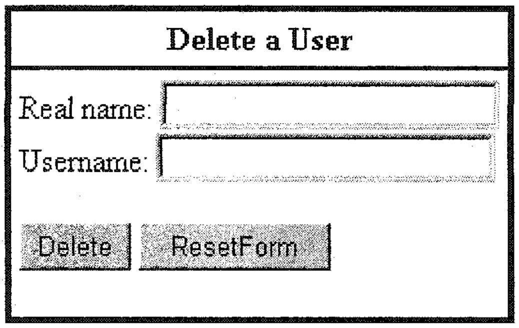


**Provider Maint. Add a User**

Real Name:
Username:
Password:
E-mail address:

For added security, a scheduled plan of activities to be performed on a recurring basis was implemented. A deletion log had been created to report all deletions. The deletion trigger log was enforced using the security features of DBMS (MySQL).

Figure 22. Remove a user dialog screen.

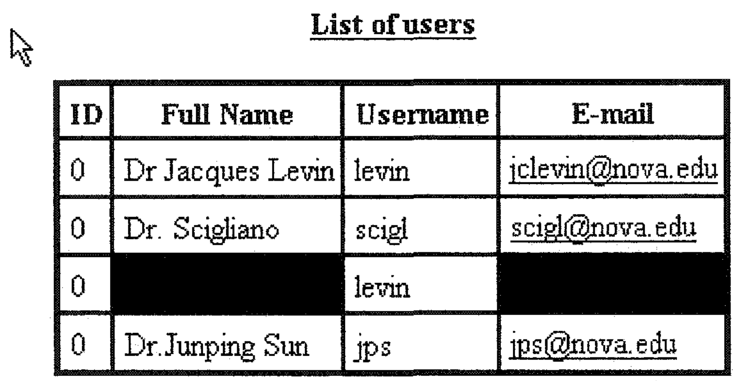


**Delete a User**

Real name:
Username:

Similarly, providers could list users in relation to their identification (ID). Lists of users could be printed as shown in Figure 23. The ID of 0 was administrative and had access privileges to all views in the application.

Figure 21. List of users.



**List of users**

| ID | Full Name | Username | E-mail |
|---|---|---|---|
| 0 | Dr Jacques Levin | levin | jclevin@nova.edu |
| 0 | Dr. Scigliano | scigl | scigl@nova.edu |
| 0 | [redacted] | levin | [redacted] |
| 0 | Dr.Junping Sun | jps | jps@nova.edu |

## Evaluation of the Prototype's Implementation and Architecture

The evaluation criteria for the current implementation were broadly categorized as scalable, maintainable, reliable, open, and secure:

1. Scalability: identified the prototype's performance and potential bottlenecks. Scalability included the ability to scale a system to problem needs, contractual requirements, budgetary constrains, and business goals and objectives (Laitinen, Fayad, & Ward, 2000).
    - Web server maintained site's availability and scaled beyond current configured parameters.

- Satisfied data transmission and scaled to accommodate more users.
- Used open architecture and industry standard communication protocols (TCP/IP and Ethernet).
- Underlying infrastructure allowed for a high degree of scalability (hardware, operating system and database management system (DBMS)).

2. Maintainability was the prototype's complexity of maintaining and configuring the system; backup and recovery strategies were evaluated. The assurance of database integrity involved redundancy in terms of data storage and processing (Cooper, et al., 1995).
    - Centralized configuration simplified backup and recovery.
    - Independent of other machines because of the centralized server.
    - Ease of programming maintained using open-source PHP.
    - Used open-source administration and monitoring tools.
3. Reliability addressed the prototype's quality, accuracy, and availability of data. In safety-critical application, programs and documentation must conform to standards defined by system reviewers. The reviewers must understand the application and be able to predict how the system will function in a particular environment (Parnas, Schouwen, & Kwan, 1990) such as medical.
    - Reliable data transmission over the Internet.
    - Multiple paths used to overcome single point path because of the Internet.
    - Robust technology based on MySQL database, ensuring consistency, integrity, and accuracy of data.
4. Open-architecture addressed interoperability, extensibility, vendor independence, and technology evolutions. Open-systems have the ability to use standard

interfaces that leverages the best of practices and software technologies from multiple vendors (Dempsey, et al., 2002).

- Industry standard operating system (Linux), web server (Apache).
- Used low cost open-source solutions.

5. Security dealt with permissions, privileges, and audit trails. Security threats have become more complex. The best way to defend against attacks involved the combination of various security products (Escamilla, 1998).
    - Included encryption of sensitive data and audit trail of any changes (data, configuration to the system).
    - Operating system, Linux, and RDBMS (MySQL) had fine grain access controls.

The prototype was designed and coded with availability and reliability in mind. The application was built with open-source administration tools. Performance tuning included setting specific flags and options on the database, the operating system, and the application code to achieve peak performance. Following the construction and tuning efforts, quality assurance was tested to measure the application's performance prior to deployment to establish acceptable quality benchmarks. All of the anticipated efforts performed well. To determine whether the Web site was operating within established operating parameters, a review of statistics generated by Web server monitoring and logging programs were perfect.

**Evaluation of the Prototype**

The investigator to evaluate the general value of the prototype conducted an evaluation. Alternative evaluation approaches were conducted to evaluate the general worth of the information system (Boloix & Robillard, 1995). Three architecture choices for storing medical data were identified. First was local at each provider's office, second, at several regional sites, and third, at a centralized Web location. In this study, the alternatives are shown in Table 9.

The centralized storage alternative was used. The centralized location for the medical data used file servers to store and process data. Personal computer clients at each client location provided the user interface and data access over telephone lines or the Internet. Architectural decisions were important as to whether a given criterion was significant or supplementary. Blaha & Premerlani (1998) recommended that special care was required for each criterion whether it was a "must" or a "want," with numerical weight assigned to each decision criteria. The investigator assumed that the evaluation reflected the diversity of experiences and perspectives found in the user community and in the healthcare industry in general.

The investigator assigned the following weights to the decision criteria. The investigator treated the criteria as 'wants' for the purpose of open evaluation to avoid spurious errors. Variables such as integrity, accuracy, and legal compliance of medical privacy rules were very important and were regarded as "musts." According to Bockle, et al., (1996), computer-system evaluation sought to answer particular questions about the system, such as its performance or availability. The following decision criterions were identified to determine success or failure; weights were assigned from a high of 100 to a low of 50:

## 1. Evaluation criteria

*Performance and ease of use*: The system must be responsive and easy to use or the providers and HMO's might ignore it. If the providers do not use the system, the stakeholders are vulnerable to problems including integrity, accuracy, and legal compliance (High weight).

*Integrity and accuracy*: Medical issues could be life and death matters. Best efforts must be deployed to ensure that the system holds correct and up-to-date data (High weight).

*Development effort*: The development effort was extremely important, but for the use of open-source tools exclusively, the cost of the development outside labor cost was not substantial (Medium weight).

*Extensibility*: As time elapses, additional uses of the medical system may be identified (Medium weight).

*Ease of maintenance*: The database will require regular update to accommodate ongoing changes in the medical industry (High weight).

*Legal compliance*: Medical laws are strictly enforced for the benefit of the patients and to curtail abuse. The systems must comply with privacy laws (High weight).

## 2. Storage alternatives

The following architectural choices were used in the evaluation of storage alternatives:

*Local storage*: Assigned a high score (100) for performance and ease of use, as all data are stored locally and there is no network or communication load to contend with. Integrity and accuracy as well as legal compliance are assigned low scores (50). The low score is because the multiple data distribution may become inconsistent with the master copy. Local storage may also require less development and maintenance effort, because there are no network and communication issues. Extensibility will be easy, as new locations could be cloned.

*Geographical storage*: A low score was assigned because users could experience communication difficulties. Geographical storage may make it easier to maintain data consistency of the master data than local storage. Medium weights for integrity, accuracy, and legal compliance were assigned because complexity from communications may increase maintenance effort. Extensibility may not be a problem as other sites could be adopted easily.

*Central storage*: Assigned the same scores as geographical storage, except that a single copy of the database can be maintained that would facilitate integrity, accuracy, and legal compliance. The best choice, based on the metrics, is central storage for the online medical collaboration, because having a single copy to update improves the odds that each site may access the correct data. It is also possible to cache data locally to help improve performance and availability.

Table 10 is a matrix of architectural choices that was used to rate the three approaches. The scores had been converted from high and medium weight to numeric values of 100 and 50 respectfully. Each decision criteria was assigned a weight of 100 or 50. A weight of 100 meant the criteria was extremely necessary and a weight of 50 meant the criteria was of medium importance. Geographical storage approach was clearly the inferior approach with a total score of 4800, followed by the local storage approach with a total score of 4600.

Table 10: Evaluation Breakdown Matrix

| **Criteria** | **Weight** | **Local Storage** | **Geographical storage** | **Central storage** |
|---|---|---|---|---|
| **Performance and ease of use** | 100 | 100 | 50 | 50 |
| **Integrity and accuracy** | 100 | 50 | 80 | 100 |
| **Development effort** | 50 | 100 | 50 | 50 |
| **Extensibility** | 50 | 100 | 100 | 100 |
| **Ease of maintenance** | 50 | 100 | 80 | 80 |
| **Legal compliance** | 100 | 50 | 80 | 100 |
| **Cost of development tools** | 100 | 60 | 60 | 100 |
| **Total Score** | | **5600** | **4800** | **5800** |

Central storage had the highest score, a total of 5800 points. The primary advantage of central storage was that there was a single copy of the database to keep updated, increasing the odds that each stakeholder could access only correct data. The central storage approach also reduced the risk of widely publishing important patient information. The central storage also facilitated integrity, extensibility, accuracy, and legal compliance. Another important benefit of centralized storage was that it reduced storage management cost. As part of architecting a system, a decision must be made about where data was stored as well as how it was stored (Blaha, 2001).

## Research Questions and Findings

The discussion under each question presents the main findings following the details of the question, its relevance, and the methods for answering the research question. The first question answered the issue of effectiveness of open-source tools in the development of secured online application. The remaining questions involved the management of sessions and profiles as compared to commercial tools, the robustness of MySQL compared to commercial databases such as Oracle, and the cost consideration of open-source over commercial tools. The last research question examined the extent the online medical prototype improved maintenance and ongoing support cost.

### *How effective are open-source tools (PHP, MySQL, Apache Web Server, Linux Operating System) compared to commercial tools (Oracle, Windows Operating System) when used exclusively to develop a secured application for online medical collaboration?*

Open-source tools such as PHP, MySQL, Apache Web server, and the Linux operating system are as effective as commercial tools in developing secured applications for all computing environments. The online prototype used open-source tools to configure and deny unauthorized access as shown in Figure 9. The object-oriented methodology explained in Chapter 3 was driven by user needs and produced the usable designed processes of the online medical prototype.

The methodology supports inter-process configuration of open-source tools as shown in Figure 10. The component interactions in Figure 11 supports the ability of open-source to react to changing needs and adapt to various environments such as the online medical prototype.

In this study, the investigator used exclusively open-source tools in the development of the online medical system. The prototype developed in this study met the requirements for a secure application. The prototype indicated that the three-tier architecture has the ability to scale to multiple users. The experience and review of the literature on open-source deployment and acceptance had shown that the open-source growth was practical and had supported disparate computing environments. The review of literature supported the investigator's assertion that open-source was winning hearts, and was a secure and maturing paradigm that continued to provide priceless alternatives to commercial tools.

*How effective are open-source tools in managing session control and administrative profiles compared to commercial tools?*

In addition to the high level of controlling sessions in the prototype, the open-source database MySQL showed a high propensity to control sessions in a consistent manner that was equal to commercial databases such as Oracle. The investigator also found that many open-source tools could be used in association with a database to secure access. Figure 17 shows the general access menu that is used to access the prototype.

The security authenticated at both the operating systems level and the database table level. Administrative profiles managed through a two level login to prevented accidental password exposure as shown in Figures 18 and 19. The prototype used the database sessions control as the basis for controlling access. The session management of the database (MySQL) was adequate and effective in controlling sessions, and was comparable to commercial database tools.

*How effective is an open-source MySQL database in providing access control compared to commercial databases over the Internet?*

MySQL provided mechanisms for security and control. The online application worked in conjunction with the Linux operating system and the DBMS to augment security provided by user names and passwords. Companies must select the combination of controls that will make a system structure work and meet the strategic goals of the company (Mintzberg, 1973). In the context of the online medical prototype, security was a proactive process that supported the balance between information integrity and access control as shown in Figure 20.

*What is the cost advantage of open-source development techniques on the Internet compared to commercial development tools?*

The business adage that "nothing good is free" fails in the open-source community's mentality. The community's mindset is based on collaboration and sharing of knowledge that furthers innovation for the good of society. The community believed that sharing is the basis of all that life was about. The result of this study supported the notion of open-source as a cost-effective business solution for producing software applications (Cohen, 2001). The use of open source exclusively eliminated product cost and reduced flexibility of support cost that could be negotiated with alternative vendors supporting open-source tools. The cost of ownership scenario shown in Appendix B supported the investigator assumption of lower ownership cost. The cost savings of open-source supported the collaborative power of emerging business critical open-source tools such as the Linux operating system. The production of high quality software and tools by the opens-source community, based on this study had generated the need for support services as a byproduct. Open-source had an extensive leverage on the best of practices (Dempsey, et al., 2002). The support need had nurtured business opportunities for the open-source members as part of the supply and demand circle in the business environment. There exists a natural tendency in humans to draw upon their own personal values, background, and beliefs in choosing strategies that can shape society (Thompson & Strickland, 1984). The open-source development paradigm, with its greater reach and willing volunteers participation through the Internet had emerged a force in the software industry. The review of literature had supported the investigator's contention that the open-source paradigm was a cost-effective method to produce high quality applications. Available tools, including the most adapted operating system, Linux and the most used web server Apache are cases in point. The software had saved high percentages of many companies' information technologies budgets. An example is E-Trade, an online trading company that saved $65millions, replacing the company's Solaris servers with Linux based servers (Galli, 2002).

*To what extent will the open-source medical online development prototype improve maintenance and ongoing support cost?*

The review of literature supported the assertion that open-source tools are superior because of their exposure to many computer science experts (Neumann, 1999). Open-source had been accepted as the best methodology for building better software and increasing economic assets of businesses (O'Reilly, 1999). The maintenance of open-source tools cost less because of ongoing quality testing the tools and software continue to endure from the open-source community (Neumann, 1999). Changes are instant and knowledge is shared with the community about improvements and preventive measures. Open-source development lowers ongoing support costs because of the many alternatives within the open-source community in solving problems. Since the development is not proprietary to a particular company, support costs can be negotiated. Established companies including International Business Machines (IBM), Electronic Data System (EDS), and many others are now moving into the open-source environments at high speed with various support offerings. IBM had spent over $1 billion on Linux software (Leavett, 2001). Big names such as Silicon Graphics and Sun Microsystems had recently made available the source code of their versions of the Unix operating system to the open-source community (Shankland, 1999). The transparency of open-source in terms of peer review and the benefit of not controlled by any single organization supported the benefits of lower cost of ownership. The researcher, based on this study, believed open-source was a sound policy making choice for cost-effective development and support. The open-source model offered companies' lower development cost, highly stable systems, and efficient security compared to proprietary systems (Leavitt, 2001).

## Cost of Ownership

Cost containment and revenues generation are some of the primary factors for business success. The open-source model answers questions that business leaders care about. Does it support the corporate vision? Does it generate competitive advantages? Does it improve overall bottom line? To answer the questions, corporate information system is one of the primary strategic factors in support activities of a corporate value chain. To achieve a distinctive competence in information systems, an entity must exploit an information-based competitive advantage. Information-based advantages require evaluating existing processes, systematic determination of how information technology can impact the entity's value chain and competition, and finally, a plan to strategically invest in information technology to achieve information-based competitive advantages (Porter, 1985). Open-source supports most corporate visions. Open-source deployment could reduce cost of an information technology infrastructure. As the information age matures, open-source had allowed companies to leverage free source code to produce software applications that offered significant competitive advantages in cost and quality. Open-source puts the organization in the driver's seat, dramatically reducing the cost of ownership and providing high reliability (Raymond, 1999). The cost of ownership for open-source compared to commercial tools and software were calculated in Appendix B, using a hypothetical company, ABC Inc.

## Summary

In this chapter, the investigator demonstrated the actual secured online medical application prototype using the object modeling methodology and exclusively open-source software and tools. Open-source languages such as PHP, Perl, and XML are becoming accepted tools in the computer industry. The chapter started with the benefits of open-source driven system development. Several security configuration techniques were discussed including complimentary tools such as Tripwire. The prototype's implementation involved authentication techniques that validated user's right to gain access to the application. The structure of the application and snapshots of the users and access screens were presented. The prototypes' objects were designed to operate in an open environment such as the Internet. The database schema and the business concept model were presented, followed by an evaluation of the prototype's implementation and architecture. The evaluation concluded with a matrix that reviewed various decision criteria including performance, extensibility and legal compliance. The matrix related the decision criteria with three storage choices. The best solution was centralized storage. Industrial testimonies supported the open-source paradigm and its future. As an example, the state department of Rhode Island developed a regulations database using open-source software. It was developed using Apache Web server software running under Red Hat Linux, and the a Dell PowerEdge server that came with a MySQL database pre-installed. There was extreme resistance and skepticism about the project. However, after delivering a fast, effective solution that came in well under budget and ahead of schedule, resistance quickly turned into strong support, and the open source approach had been used in other strategic initiatives in the department (MySQL, 2002). Westone, a global manufacturer of medical products, had to carefully manage data about its many customer constituencies, including doctors, hospitals, patients, and other health professionals. After testing Oracle Database and MySQL, IT Director, Cal Pearson noted, " I have been testing various databases to replace our old system for about three years. Once I brought MySQL into our computing lab for evaluation, I knew we had found a long term solution." (MySQL, 2003). Critical to the success of this study, the investigator had proven the power of open-source as the overall success of the prototype's implementation. The online medical application proved that the potential impact of open-source tools and software on computing resources in the future would be substantial. The power and growth of open-source could be ascertained from a sentence of Microsoft Corporation's recent 10-Q statement filings with the Securities Exchange Commission on the possible effect of open-source according to Guterman (2003), "To the extent the open-source model gains increasing market acceptance, sales of Microsoft's products may decline, the company may have to reduce the prices it charges for its products, and revenues and operating margins may consequently decline" (p. 1). In conclusion, the open-source software development paradigm, that exposes source-programming code for free to be inspect, test, and alter to meet specific requirements. The paradigm is enjoying so much success in corporate America that even Microsoft had taken notice (MySQL, 2003).

# Chapter 5

Conclusions, Implications, Recommendations, and Summary

## Overview

Designing and implementing a secured application is aimed at meeting the business objectives of providing confidentiality, availability, and access for legitimate users of the networked resources (White, Fisch, & Pooch, 1996). Internet security is not that different from single host security in terms of goal. In this chapter, the investigator reviewed the results of the study and the prototype that utilized the object-oriented development methodology. White, et al. (1996) indicated that there were basic actions to secure a system from all but the most talent and persistent of intruder. The chapter closes with the investigator's conclusions, implications, recommendations, and a summary of the study.

Managing complex technology in a dynamic computing environment such as the Internet required the use of secure open-source tools and applications. The level of information quality in current open-source applications is extensive because of evolving peer review and collaboration globally. Linux had been developed through a community-based process using the Internet and open to anyone with the right technical skills who was willing to participate (Dempsey et al., 2002).

## Conclusions

### *Conclusion 1: Open-source tools*

The primary outcome of this study was a cost-effective enterprise-wide open-source medical solution that met requirements for managing, operating, and supporting an online secure application. Analysis of the prototype indicated that open-source tools and software provided high quality and optimal security solutions comparable to commercial tools and software. Open-source tools such as PHP, MySQL, XML, and LINUX are among the best development tools for building dynamic three-tier systems. Open-source tools offer developers a cost effective, robust, and an efficient way to build quality applications (Medinets, 2000). Open-source driven applications increase the economic assets of companies (Neumann, 1999; Mockus, et al., 2000).

### *Conclusion 2: The Object-oriented design methodology*

The online medical prototype resulting from this study delivered a portal presentation that allowed secure access to medical data. The prototype utilized the object-oriented design methodology that produced a flexible and robust data model. The object-oriented design methodology is acknowledged as having the best benefits in the successful development of complex systems (Johnson, 2000). The object-oriented development methodology offered exceptional implementation solutions when associated with prototyping in the development process. The waterfall approach is not recommended for application development with substantial uncertainties in requirements. According to Boehm (1981), a clear statement of requirements was necessary for the traditional model to be effective. The traditional approach of development fails in comparison to prototyping (Cerpa, 1999). The investigator, based on the review of literature, believed that the traditional or waterfall model might have failed in comparison to prototyping because it takes a long time to complete and omits most application requirements.

Open-source security and environmental tools were used to compliment the standard Linux security. Tripwire tracked changes to files and reports exceptional changes in a report that alerts systems administrators. Sudo limited authorized users access to a subset of commands, files, and hosts on the network. The configuration file defined aliases for users, hosts, and commands that denotes who had access to what commands and on which hosts. Secure shell (ssh), a tool for remote access to system resources, used public-key cryptography to establish a secure channel of communication over public networks such as the Internet.

*Conclusion 3: Evaluation of the online medical prototype*

The evaluation of the prototype indicated that it was scalable, maintainable, and reliable. The study also addressed the prototype's quality, accuracy, and availability of data. As exclusively open-source, the prototype addressed the investigator's resolve for interoperability, extensibility, and vendor independence. In general, results of the study suggested that the computing public had adopting open-source effectively and were actively involved in the promotion of the open-source paradigms on jobs and organizations. The findings also indicated that the dynamic nature of information technology might force organizations to find ways to cope with constant skill shift, proactive training requirements, and technological environment scanning for cost-effective tools that could reduce overall information technology costs.

Results of the study further indicated that cost, quality, and shared support by online users and commercial entities were factors for users identifying with many open-source operating systems, databases, web servers, and programming languages in the computing environments. Open-source languages such PHP, Perl, and XML are becoming accepted tools in the computer industry (Medinets, 2000). As users and developers continue to embrace the phenomenal growth of open-source, over time, sufficient confidence will emerge to implement innovative and cost-effective applications in industry. Consequently, from security, cost, and maintenance standpoints, the study supports the assumption that open-source tools can be used to develop secure architecture for collaboration in any computing environment.

## Implications

Security analysis depends on assumptions, and no one system is totally secure. It is difficult to exert control over a heterogeneous network such as the Internet. Security risk could be understated or overstated. Measures of security in this study are sometimes subjective and not certainties. As an example, the flawed Patriot missile was presumed to work under a much more friendly environment than that of the Arabian Desert during the Operation Desert Storm in 1991. Later analysis after the fact downgraded its effectiveness from 95% to 13% (Neumann, 1993). Experience with network bottlenecks had shown that properties of segmented network links could have a significant impact on the performance of the TCP/IP protocol that may be hard to investigate. The performance of connections along multiple paths and links such as the Internet are of special concern to the networking community (Cooper, et al., 1995). Connection from clients to the online medical system may experience slow traffic and response because of slow links at the point of origination.

### *Implication 1: Securing a system in the context of the Internet*

It is implied that the effects of online information disaster could be greater as all the records for a particular participating client could be destroyed or lost. Online systems could become susceptible to computer viruses that could spread from system to system, clogging computer memory and destroying programs and data. The systems are susceptible because they could be accessed at many points in the Internet network. The Internet had been less collegial and trustworthy compared to a secured intranet environment (Oppliger, 1997).

The implication of internal control and safeguarding of information systems are problematic. Unauthorized users pose the biggest threat to security; most historical security breaches and damage came from insiders, not outside intruders. Controls that are effective but do not prevent authorized users tend to be cumbersome. Requiring too many passwords could weaken usage or go unused. Network attacks in today's online environment are a serious problem and difficult to defend (Rieken & Weiman, 1992).

### *Implication 2: Implications for future research*

Open-source adoptions and the influence on software development using prototyping had been disseminated to help practitioners improve their processes and reduce information systems infrastructure cost. Results of this study should shed light on the preferences and the future of the competition between open-source tools and traditional proprietary products in the computing environment. The investigator assumed that open-source applications including Linux, Apache, XML, and MySQL were becoming superior products of choice, with the Internet serving as the communication tool for global collaborations. Open-source projects had revealed the deficiencies of traditional models that relied on face-to-face collaboration. The open-source model had proven through many projects that the best results could be obtained in team efforts using the Internet (Schach, 1998). The open-source community, and the nature of its contribution to the world of computing, is little known and poorly understood by society in general. Over time, continued acknowledgments could lead to greater acceptance.

The results of this study should broaden the view that the emerging open-source paradigm could change the way the computing industry works. Businesses had incorporated open-source codes with their proprietary commercial software without the threat of anti-trust lawsuits (Railsback, 2001). The open-source model had given the online community the general capacity to promote and further the ideals of democracy and decentralization of technological knowledge of emerging computing trends. In sluggish economic periods, open-source could be a tremendous help to businesses struggling to maintain survivable budgets (Perlow, 2001; Railsback, 2001).

Studies had shown that businesses were adopting the open-source model and tools to augment internal security (Mockus, Feilding & Herbsleb, 2000). Sonnenreich & Yates (2000) stated that, the benefits of open-source operating systems like Linux and OpenBSD were excellent and offered a cost-effective model for building intrusion-proof firewalls for businesses. The open-source model could affect the strategic directions of many businesses. Technological advances and the quality of global collaborative efforts in software development could continue to make open-source the life-blood of global cooperation in application development and information technology. Proponents of the model believed that leveraging the Internet and the open-source model were an alternative, economic, and rewarding way of producing robust software that could reshape the multi-billion dollar commercial software industry (Dempsey, Weiss, Jones & Greenburg, 2002).

Linux had been the fastest-growing platform for all business sizes. As supported by Binstock (2002), Linux historians noted at the Linux World trade show on August 12, 2002, the acceptance of Linux. Linux was a globally viewed as a powerful operating system by the positive events at the trade show. Keynote addresses by Sun Corporation's Scott McNealy and Oracle's Larry Edison established that Linux was here to stay and destined for the enterprise-computing environment. Adding to the open-source mystique was the presence of a Microsoft booth at the fair. There were server vendors on the show floor, including Dell, Hewlett-Packard, and IBM, offering large high-end server boxes for the enterprise. The most radical occurrence cited by the author was a demonstration to the City of San Francisco on a proposed measure requiring the purchase of only open-source software (Binsock, 2002).

The findings of this study should be of practical value to medical organizations that have discovered the technical difficulties of balancing privacy issues with cost-effective Web-based collaborative applications to share knowledge and deliver services to their constituents. Building on the research, the healthcare industry can enhance technical data warehousing capabilities and make them available to practitioners on the Internet for use by healthcare decision makers. The healthcare industry had been decentralized with large autonomous collections of data, gripped with questions of quality and integrity (Berndt, Fisher, Hevner & Studnicki, 2001).

Results of the study could help developers select an architecture based on the outcome of object modeling techniques that uniquely addressed their application needs. The Internet had created a conflict of choices on emerging technologies and architectural decisions that could be important to companies in developing more complex processes. The object-oriented modeling technique could resolve most of the architectural issues of companies and help improve their process dimensions. The object-oriented model, in this study, coupled with prototyping and open-source tools had answered the questions of business process dimensions such as process models, phases, roles, activities, and object artifacts. The models emphasized in this study had been well tested in real projects in several organizations and have evolved into the most respected design for complex projects (Henderson-Sellers & Unhelka, 2000). The merger with the open-source model in this study had reinforced the combination suitable for best software quality development at a sustainable budget.

The research questions posed in the study were answered in the research findings and report. Results of this study should show how open-source became a practical force in all segments of the computing environment. The study used open-source tools and object-oriented design methods in developing a secure prototype that was cost-effective and easy to maintain. Results of this study could raise new issues about software development when the Internet is used as the communication medium. The implications of this study may serve to encourage further studies of the open-source model. The findings of this study may generate possible scholarly publications on the values of the open-source model and the use of object-oriented modeling to produce highly complex applications on the Web. The research, based on the prototype produced, could motivate medical organizations to protect patients' privacy, a task that could be achieved using open-source security tools. The use of open-source tools could minimize the cost of such implementation to the medical organization. Internet security is difficult to maintain, but some open-source tools contain advantages, because hackers, computer scientist, and people of goodwill to society populate the open-source community.

Results of this study should allow the reader and other investigators to evaluate the importance of user involvement in application development. The importance of users in application development is not new, and object-modeling techniques make users the center of the development effort. The objective of the object-oriented techniques is to establish artifacts and understand the behavior of objects. Users are closely involved in real processes and could help in modeling "use cases" and interfaces. The use of three-tier development could be an added benefit that could help commercial developers to reduce cost by using a better design methodology and free development tools from the open-source community.

## Recommendations

Open-source applications tend to be much lower in cost than the proprietary equivalent. Use of open-source can reduce development cost and increase innovation. Open-source, from this study, are commercially supported, widely used, and have curved a proven records of security and reliability. Use of open-source tools can diversify an organizations cost and security risk of dependence on commercial software and tools.

### *Recommendation 1: Future Research and Development*

It is recommended that a study in the context of computer be conducted in the areas of individual values of hacking open-source code onto commercial products. Studies have addressed software pirating and individual values about illegal software coping. These studies had been conducted within the scope of business ethics and human relations (Taylor & Shim, 1993; Swinyard, Rinne & Keng Kau, 1990; Weber, 1993). A study in the context of computer science had not been conducted about individual values, free code sourcing, and hacking open-source code onto commercial tools and applications. Object-oriented development has yet to measure the productivity of the process against competing software development environments in a scientific setting. An objective investigation is recommended that could uncover the dilemma of today's software industry. As Ledgard (2001) stated, "An industry with no scientific measure of productivity is like an emperor without an empire" (p. 128).

Another future study is recommended for teaching strategies outside the old school of procedural programming and the development of textbooks that could encourage object oriented thinking. Westfall (2001) argued that academia is deficient in teaching object-oriented classes, stating that "I reflected on my own personal experiences with object-oriented programming; the texts were not really communicating the concepts, as they were written for people whose backgrounds were in procedural languages. I was not learning 'object-thinking,' because the authors were not writing from that perspective" (p.131). Hamada & Scott (2001) indicated that a number of studies about how professionals acquire knowledge supports the notion that learning is inextricably intertwined with multidirectional activities such as work and play, and in fact learning is a social activity.

Their position about learning affirms that the process of knowledge acquisition cannot be abridged from the process of knowledge application. According to the authors, knowledge relative to acquiring and applying is temporary, developmental, and socially and culturally mediated. Thus, learning is a nonobjective entity (Fosnot, 1992).
Further study is also recommended on how open-source, in particular web-based applications that have evolved in the last decade, adds value to information technology. Added value could also be induced by looking at the cost arising from web-based efficiency from many quantifiable domains, such as management and licensing from older models, in relation to collaboration tools.

*Recommendation 2: Contribution to the Field of Study and Advancement of Knowledge*

It is recommended that the results of this research have shown the means and related features for designing web-based applications using free open-source tools. The features had supported and delineated aspects of the social context of open-source in a capital environment that had contributed to knowledge. It is evident that the social context had motivated the open-source community, improved the communication authority of the Internet, and framed the views and social norms of the collaboration paradigm online. As established in the study, the Internet afforded greater opportunities for global participation in producing applications that are superior, in some cases, to their commercial or proprietary counterparts. The open-source revolution had helped broadened the concept of equitable in terms of participation. It had also and reduced most of the pervasive social ills, including gender and race discrimination. The study challenged and indeed put into question most of the traditional models of management controls and team activities. The results showed that the Internet's anonymity and other social norms for online participation and collaboration had increased overall knowledge integration, broadened the repertoire of scientific knowledge, and invoked thoughts that previous traditional models had failed to provide.

Prior research and opinions about the open-source revolution within the modern computing landscape had suggested charges in corporate attitudes and greed. Open-source had opened a social outlet outside the profit motive (Cohen, 2001). The research had shown how valuable applications could be made available to people by the open-source community regardless of income and country. From the theoretical view, open-source technology and development could impact the economy as technological innovations take the center of the economic capacity of nations. It is expected that open-source use could drive the production of free and innovating products to a lower cost. The intrinsic impact could be the faster growth in terms of output and employment outside the traditional, dominant, multinational companies, and could socially redistribute wealth in a mega information paradigm. The effects of open-source adoption on employment, production process, national economies, and the world scene could be substantial.

## Summary

The Internet brought the online community a lot of information and convenience. Security remains a problem to online users. With the advent of the Internet, the issues of managing security online became important. The problem addressed in this study involved the security of sensitive medical data and strategies for online software development. The significance and problems of software development using exclusively open-source tools and software were determined. The anonymity of the Internet created generous opportunities for misbehavior by users. The Internet lacked a social system to create and promote trust outside the open-source community. The field of security and software development strategies had changed. Closed-source proprietary software had been considered the lifeblood of computer systems; however, there are associated risks with proprietary software. Building reliable, secure systems on the Internet remained elusive (Weaver, Vetter, Whinston & Swigger, 2000). Ahn, Sandhu, Kang, & Park (2000) observed that most existing Web-based workflow systems provided minimal security services such as authentication of users. Security remained the most important concern in the growing open-source community. New security threats are developed every day, thereby threatening Web systems all over the world. Online systems are historically security-sensitive because of the dynamic nature of the Internet.

The goal of the investigator in this study was to design a secure system for medical collaboration on the Internet with multiple access security using exclusively open-source tools. The online medical management prototype provided online collaboration, enabled effective data communication, and allowed access security that used open-source tools for implementation. The investigator examined how open-source tools could be structured to the needs of a three-tier development over the Internet. Open-source tools such as Linux, an operating system; MySQL, an object-relational database system; Personal Home Page (PHP), an application server and programming language; and Apache, a widely deployed open-source Web server, were used to develop a prototype medical online system. Supporting open-source, development advocates had stated that open-source tools such as PHP, MySQL, XML, and LINUX were the answer to building cost-effective, faster, and maintenance-free dynamic three-tier systems (Medinets, 2000). The dynamic aspect of information systems today calls for creative thought, with particular attention to quality, cost, and project control.

To achieve the goal in this study, the investigator adopted the Object Modeling Techniques (OMT) methodology to analyze, design, and build the online medical system. The analysis, design, and implementation of the online prototype were concluded with a robust and secured online application. The business concept model (Figure 10) was transformed into an object-relational data model, followed by an intensive security and architectural evaluations. The evaluations included decisions criteria involving performance, security, and storage choices. Sensitive medical data must be protected from unauthorized access. The Internet is the most efficient medium to access the decentralized nature of medical data (Huston & Huston, 2000). Complimentary security tools are necessary to protect medical data. Open-source adoptions and their influence on software development using prototyping could be disseminated to help practitioners improve their processes and reduce information systems infrastructure cost. The investigator believed that open-source applications including Linux, Apache, XML, and MySQL are becoming superior products of choice, with the Internet serving as the communication tool for global collaborations. Open-source projects have revealed the deficiencies of traditional models that rely on face-to-face collaboration. The open-source model has proven through many projects that best results are obtainable in team efforts using the Internet (Schach, 1998). Businesses are incorporating open-source codes onto their proprietary commercial software without the threat of anti-trust lawsuits. Studies have shown that businesses are adopting the open-source model and tools to augment internal security (Mockus, Feilding & Herbsleb, 2000). The open-source model had affected the strategic directions of many businesses. Technological advances and the quality of global collaborative efforts in software development had continued to make open-source the life-blood of global cooperation in application development and information technology.

Online collaboration had allowed computer scientists to leverage technical resources to produce software applications such as the Linux operating system. Software applications produced by the online development community have led to the open-source paradigm. The open-source paradigm is a powerful collaborative technique of shared knowledge and research to meet common computing needs (Raymond, 2001). The primary focus for the open-source community had been to develop a wide range of free information management applications and tools. Open-source applications had evolved through the years and include operating systems, databases, scripting languages, and security tools that are operational in many computing environments. The use of open-source tools in developing secured systems provides a multitude of benefits. The open-source benefits include increased security in open environments such as the Internet.

The security of open-source software had been possible because the source code is in the public view, subjecting it to greater scrutiny and instant fixes to problems. The open-source model lends its acceptance to collaboration. The open-source model had become a bill of rights for computer users. It allows users to make copies of an application, improve the application's source code, and distribute those copies. According to Perens, "Research laboratories have adopted the open-source model to share information critical to scientific investigation. There is also the benefits to companies that exist solely to support open-source applications.

Open-source scripting tools have gained vast database deployment options from commercial databases including Oracle, Sybase, and Informix. The deployment of a three-tier development process with open-source databases, scripting languages, and Web servers inside an organization had become revolutionary and economical. When commercial systems are not adequately robust and secured, designers may consider open-source components to build robust systems (Neumann, 1999). Open-source had been argued as a superior methodology for the development of end-user software that increases economic assets of companies (O'Reilly, 1999).

The investigator implemented an online medical system design using open-source tools. The prototype used the object-oriented design methodology, a systematic software development approach that used objects and notations to express the underlying business concepts (Blaha, 2002). There are great benefits to object-oriented systems development. Investigators have done objective quantification of the benefits of object-oriented design. Henderson-Sellers & Unhelka (2000) indicated that to implement a complex system successfully, one must use a development methodology like the object-oriented model.

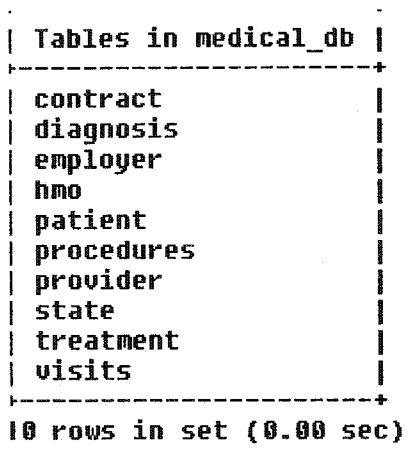
The investigator also used prototyping to build the online medical application that allowed the end-user's participation and evaluation throughout the development process (Laudon & Laudon, 1996). Most online system requirements are dynamic and may require rapid development, such as prototyping. The investigator used prototyping that was allowed flexibility of the design, helped in the functionality of system components, and made detection of problems easier. According to Cusumano & Selby (1997), many companies now use prototyping as well as concurrent design, build, and test activities to control iterations during system development. Prototyping facilitates better design choices early in the software development process (Zucconi, Mack, & Williams, 1990).

# Appendixes

# Appendix A

## Description of Database Tables Used in the Prototype

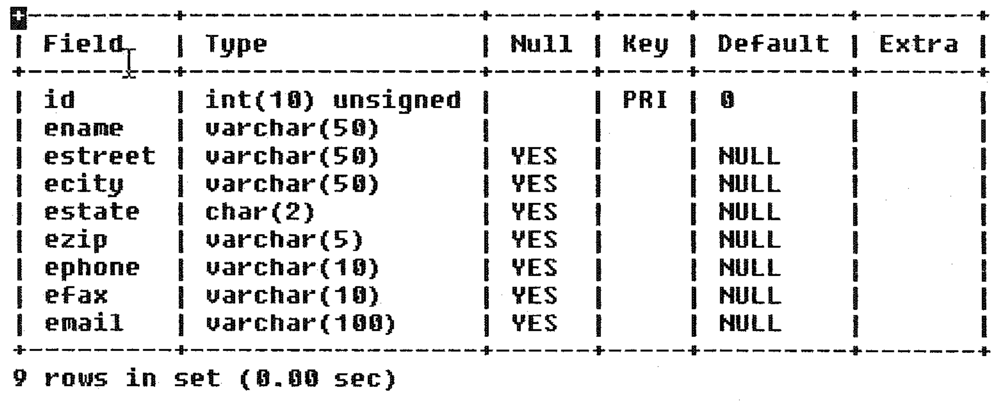
Tables of the online medical system



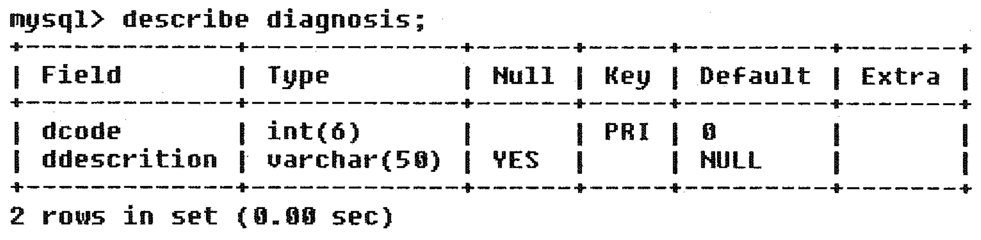
```
| Tables in medical_db |
+----------------------+
| contract             |
| diagnosis            |
| employer             |
| hmo                  |
| patient              |
| procedures           |
| provider             |
| state                |
| treatment            |
| visits               |
+----------------------+
10 rows in set (0.00 sec)
```

Attributes of the employees and diagnosis tables captured using MySQL keyword DESCRIBE. The command captures all attributes associated with the table.

**mysql> describe employers;**

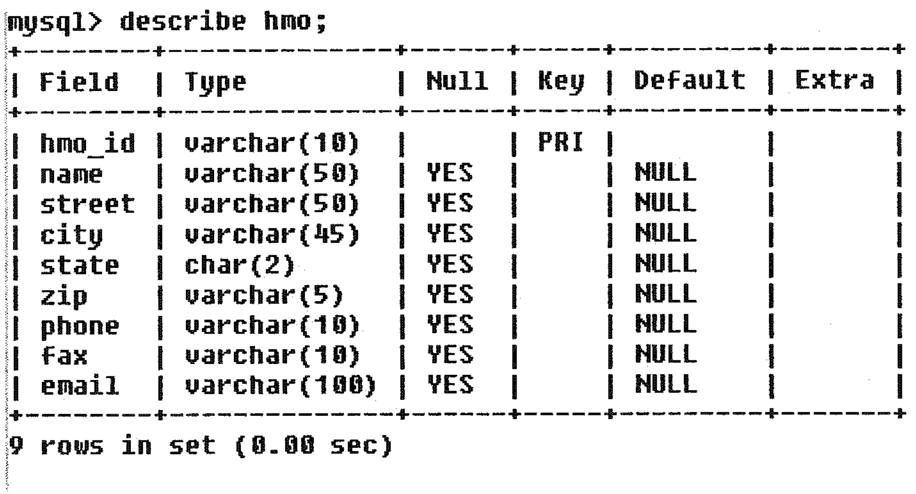


```
+---------+-------------------+------+-----+---------+-------+
| Field   | Type              | Null | Key | Default | Extra |
+---------+-------------------+------+-----+---------+-------+
| id      | int(10) unsigned  |      | PRI | 0       |       |
| ename   | varchar(50)       |      |     |         |       |
| estreet | varchar(50)       | YES  |     | NULL    |       |
| ecity   | varchar(50)       | YES  |     | NULL    |       |
| estate  | char(2)           | YES  |     | NULL    |       |
| ezip    | varchar(5)        | YES  |     | NULL    |       |
| ephone  | varchar(10)       | YES  |     | NULL    |       |
| efax    | varchar(10)       | YES  |     | NULL    |       |
| email   | varchar(100)      | YES  |     | NULL    |       |
+---------+-------------------+------+-----+---------+-------+
9 rows in set (0.00 sec)
```



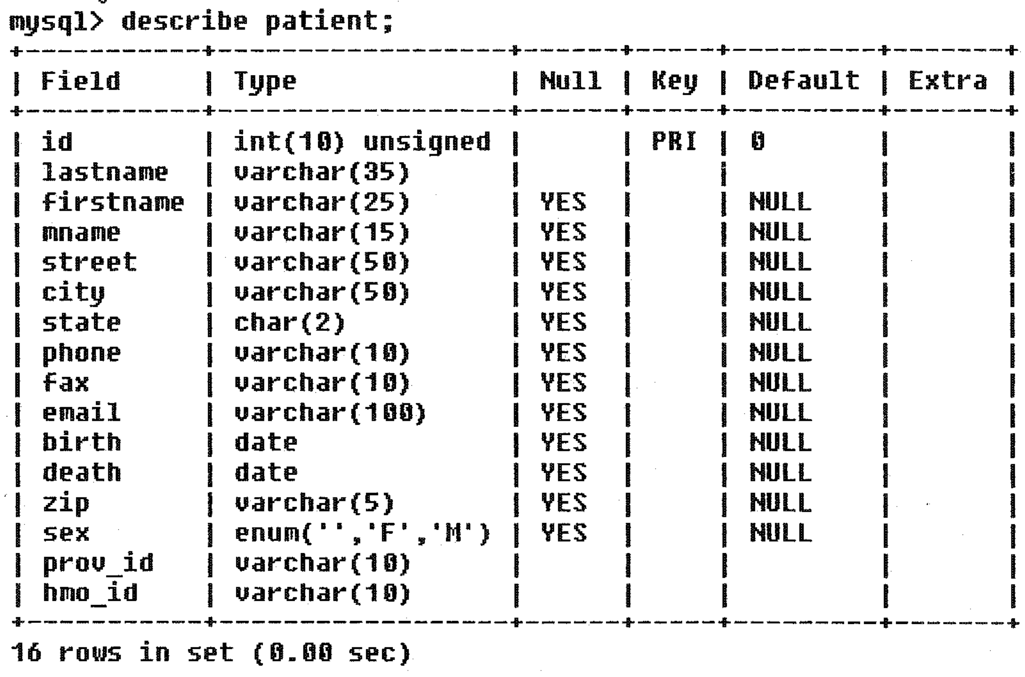
```
mysql> describe diagnosis;
+-------------+-------------+------+-----+---------+-------+
| Field       | Type        | Null | Key | Default | Extra |
+-------------+-------------+------+-----+---------+-------+
| dcode       | int(6)      |      | PRI | 0       |       |
| ddescrition | varchar(50) | YES  |     | NULL    |       |
+-------------+-------------+------+-----+---------+-------+
2 rows in set (0.00 sec)
```

The HMOs table



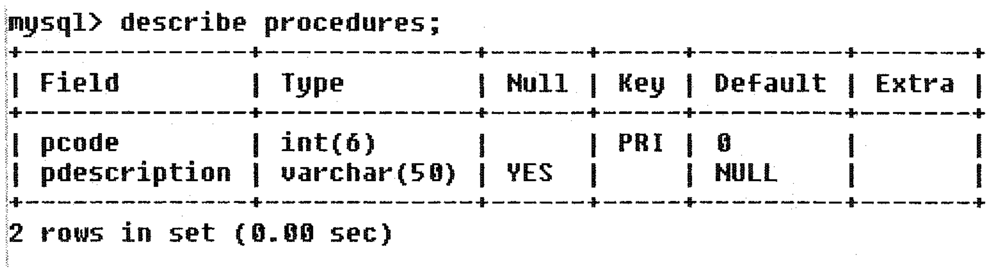
```
mysql> describe hmo;
+--------+--------------+------+-----+---------+-------+
| Field  | Type         | Null | Key | Default | Extra |
+--------+--------------+------+-----+---------+-------+
| hmo_id | varchar(10)  |      | PRI |         |       |
| name   | varchar(50)  | YES  |     | NULL    |       |
| street | varchar(50)  | YES  |     | NULL    |       |
| city   | varchar(45)  | YES  |     | NULL    |       |
| state  | char(2)      | YES  |     | NULL    |       |
| zip    | varchar(5)   | YES  |     | NULL    |       |
| phone  | varchar(10)  | YES  |     | NULL    |       |
| fax    | varchar(10)  | YES  |     | NULL    |       |
| email  | varchar(100) | YES  |     | NULL    |       |
+--------+--------------+------+-----+---------+-------+
9 rows in set (0.00 sec)
```

The patient's tables



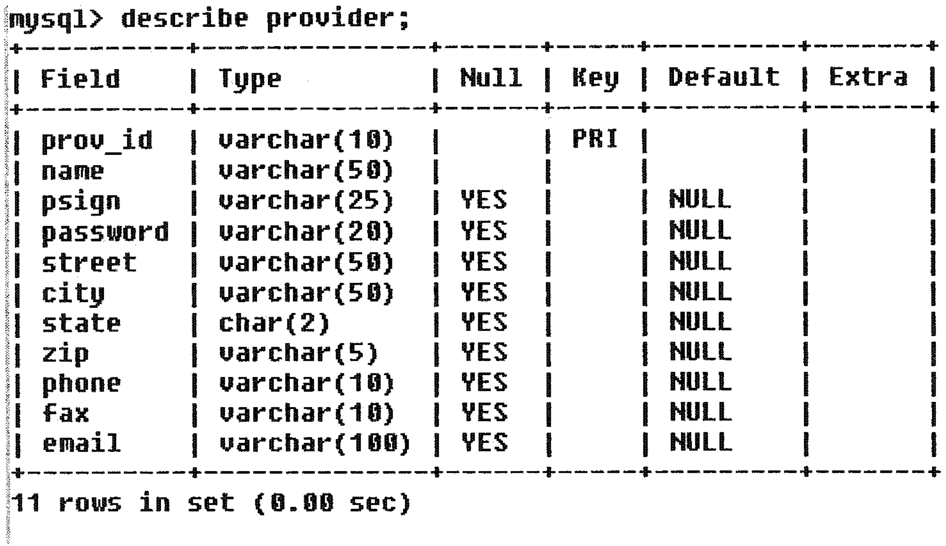
```
mysql> describe patient;
+-----------+------------------+------+-----+---------+-------+
| Field     | Type             | Null | Key | Default | Extra |
+-----------+------------------+------+-----+---------+-------+
| id        | int(10) unsigned |      | PRI | 0       |       |
| lastname  | varchar(35)      |      |     |         |       |
| firstname | varchar(25)      | YES  |     | NULL    |       |
| mname     | varchar(15)      | YES  |     | NULL    |       |
| street    | varchar(50)      | YES  |     | NULL    |       |
| city      | varchar(50)      | YES  |     | NULL    |       |
| state     | char(2)          | YES  |     | NULL    |       |
| phone     | varchar(10)      | YES  |     | NULL    |       |
| fax       | varchar(10)      | YES  |     | NULL    |       |
| email     | varchar(100)     | YES  |     | NULL    |       |
| birth     | date             | YES  |     | NULL    |       |
| death     | date             | YES  |     | NULL    |       |
| zip       | varchar(5)       | YES  |     | NULL    |       |
| sex       | enum('','F','M') | YES  |     | NULL    |       |
| prov_id   | varchar(10)      |      |     |         |       |
| hmo_id    | varchar(10)      |      |     |         |       |
+-----------+------------------+------+-----+---------+-------+
16 rows in set (0.00 sec)
```

List of attributes for the procedure table



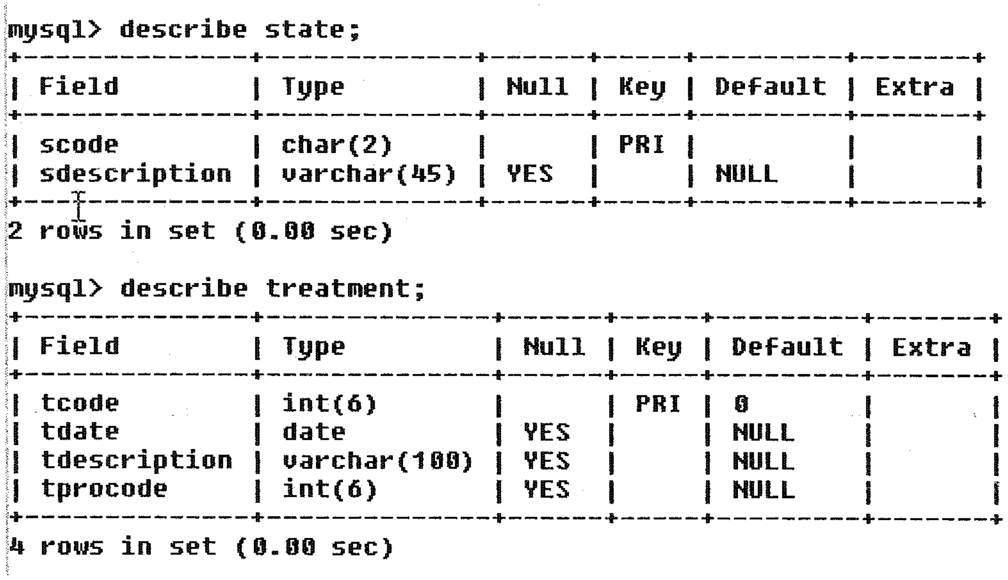
```
mysql> describe procedures;
+--------------+-------------+------+-----+---------+-------+
| Field        | Type        | Null | Key | Default | Extra |
+--------------+-------------+------+-----+---------+-------+
| pcode        | int(6)      |      | PRI | 0       |       |
| pdescription | varchar(50) | YES  |     | NULL    |       |
+--------------+-------------+------+-----+---------+-------+
2 rows in set (0.00 sec)
```

Attributes of the provider's table



```
mysql> describe provider;
+-----------+--------------+------+-----+---------+-------+
| Field     | Type         | Null | Key | Default | Extra |
+-----------+--------------+------+-----+---------+-------+
| prov_id   | varchar(10)  |      | PRI |         |       |
| name      | varchar(50)  |      |     |         |       |
| psign     | varchar(25)  | YES  |     | NULL    |       |
| password  | varchar(20)  | YES  |     | NULL    |       |
| street    | varchar(50)  | YES  |     | NULL    |       |
| city      | varchar(50)  | YES  |     | NULL    |       |
| state     | char(2)      | YES  |     | NULL    |       |
| zip       | varchar(5)   | YES  |     | NULL    |       |
| phone     | varchar(10)  | YES  |     | NULL    |       |
| fax       | varchar(10)  | YES  |     | NULL    |       |
| email     | varchar(100) | YES  |     | NULL    |       |
+-----------+--------------+------+-----+---------+-------+
11 rows in set (0.00 sec)
```

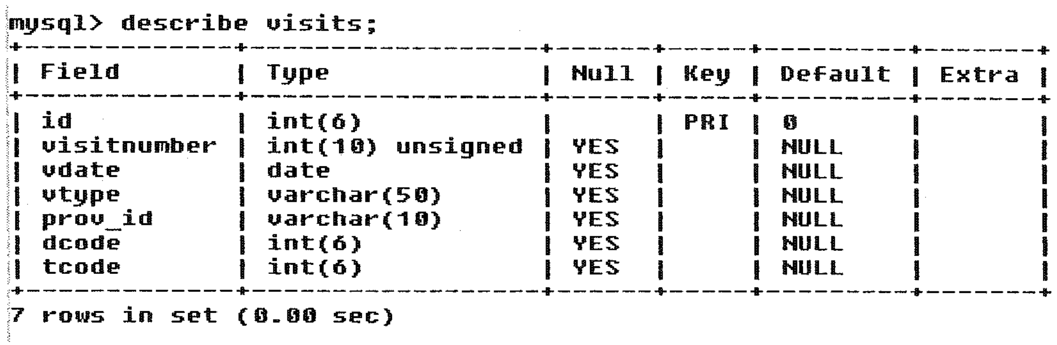
Attributes of the state and treatments tables



```
mysql> describe state;
+--------------+-------------+------+-----+---------+-------+
| Field        | Type        | Null | Key | Default | Extra |
+--------------+-------------+------+-----+---------+-------+
| scode        | char(2)     |      | PRI |         |       |
| sdescription | varchar(45) | YES  |     | NULL    |       |
+--------------+-------------+------+-----+---------+-------+
2 rows in set (0.00 sec)

mysql> describe treatment;
+--------------+--------------+------+-----+---------+-------+
| Field        | Type         | Null | Key | Default | Extra |
+--------------+--------------+------+-----+---------+-------+
| tcode        | int(6)       |      | PRI | 0       |       |
| tdate        | date         | YES  |     | NULL    |       |
| tdescription | varchar(100) | YES  |     | NULL    |       |
| tprocode     | int(6)       | YES  |     | NULL    |       |
+--------------+--------------+------+-----+---------+-------+
4 rows in set (0.00 sec)
```

Attributes of the visits table



```
mysql> describe visits;
+-------------+------------------+------+-----+---------+-------+
| Field       | Type             | Null | Key | Default | Extra |
+-------------+------------------+------+-----+---------+-------+
| id          | int(6)           |      | PRI | 0       |       |
| visitnumber | int(10) unsigned | YES  |     | NULL    |       |
| vdate       | date             | YES  |     | NULL    |       |
| vtype       | varchar(50)      | YES  |     | NULL    |       |
| prov_id     | varchar(10)      | YES  |     | NULL    |       |
| dcode       | int(6)           | YES  |     | NULL    |       |
| tcode       | int(6)           | YES  |     | NULL    |       |
+-------------+------------------+------+-----+---------+-------+
7 rows in set (0.00 sec)
```

## Appendix B

## Cost of Ownership

**ABC Inc.**
**Hypothetical Company**

Cost of Ownership

| Environment | Proprietary | Open-source | Product Cost | Support cost | Proprietary | Open-source |
|---|---|---|---|---|---|---|
| Web Server | IIS | Apache # | 20000 | 20000 | 40000 | 20000 |
| Database | Oracle | MySQL # | 60000 | 40000 | 100000 | 40000 |
| Operating System | IBM AIX | Linux # | 10000 | 10000 | 20000 | 10000 |
| Total | | | 90,000 | 70000 | 160000 | 70000 |

# = Free software or tools

**Benefits and assumptions**

Weather Services uses MySQL for real-time weather data and imagery, timeliness of data delivery, accuracy, and reliability (Open, 2001 p.25). Let's assume that support remaining the same fee of $70,000 per year for both commercial and open-source options. A mid-sized company can reduce its overall cost of software by $90,000, representing approximately 56% cost savings of its IT budget.

*Advantages Assumed*

1. No risk of losing support, as a fall over is the online community.
2. No cost to upgrade.
3. Instant security and other fixes online.

**References**

Adams, A., & Sasse, M. (1999). Users are not the enemy. *Communication of the ACM,* 42(12), 40-46.

Adler, P. (1986). New technologies, new skills. *California Management Review.* 29(1), 9-27.

Adler, P., McDonald, D., & MacDonald, F. (1992). Strategic management technical functions. *Sloan Management Review*, 33(2), 19-37.

Ahn, G., Sandhu, R., Kang, M., & Park, J. (2000, July 26). Injecting RBAC to secure a Web-based workflow system. In the *Proceeding of 3rd ACM Workshop on Role-based Access Control,* Berlin, Germany, 1-10.

Akker, T., Snell, Q. O., & Clement, M.J. (2001, May 3-4). The YGuard access control model: set-based access control. Paper. Sixth ACM Symposium on Access control models and technologies. Chantilly, VA: ACM Workshop on Role Based Access Control.

Allison, J. (2001, July 16). Which is more secured? Open Source versus Proprietary. *Interactive Week,* (8) 28, 23-26.

Alavi, M. (1984). An assessment of the prototyping approach to information systems development. *Communication of the ACM*, 27(6), 556-563.

Anido-Rifon, L., Fernandez-Iglesias. M.J., Llama-Nistal, M., Caeiro-Rodriguez, M., Santos-Gago, J., & Rodriguez-Estevez, J.S.(2001). A component model for standardized Web-based education. *ACM Journal of Educational Resources in Computing*, 1(2), 1-21.

Armoni, A. (2000). *Healthcare Information Systems*. Hershey, PA: Idea Group Publishing.

Armour, P. G. (2002). The organism and the mechanism of projects. *Communication of the ACM*, 45(5), 17-20.

Babcock, C. (2000, October). Linux boosts Unix: upstart Operating System proves friend, not foe, to Unix. *Interactive Week,* 68-70.

Barkley, J., Kuhn, R., Rosenthal, L., Skall, M., & Cincotta, A. (1998). Role-based access control for the Web. *National Institute of Standards and Technology*. Retrieved March 10, 2003, from the World Wide Web:http:/hissa.ncsl.nist.gov/rbac/cals-paper.html

Bashir, I., Serafini, E., & Wall, K. (2001, February). Securing network software applications. *Communication of the ACM*, 44(2), 29-31.

Barret, D. J. (1996). *Bandits on the Information Superhighway*. Sebastopol, CA: O'Reilly & Associates, Inc.

Behlendorf, B. (1999). *Open-source: Voices from the open-source revolution.* Retrieved March 10, 2003, from the World Wide Web: http://www.oreilly.com/catalog/opensources/book/brain.html

Bell, D., & LaPadula, L. (1973). *Secure Computer Systems: Mathematical Foundation and Model.* Bedford, MA: MITRE Corp.

Bental, D., & Cawsey, A. (2002). Personalized and adaptive systems for medical consumers applications. *Communication of the ACM*, 45(5), 62-65.

Berard, E. (1993). *Essays on Object-Oriented Software Engineering*. Englewood Cliffs, NJ: Prentice-Hall.

Berlind, D. (2002). Open-source: IBM's deadly weapon. *ZDNet.* Retrieved March 10, 2003, from the World Wide Web: http://techupdate.zdnet.com/techupdate/stories/main/0,14179,2860394,00.html

Berghel, H. (1994). New wave prototyping: use and abuse of vacuous prototypes. *Interactions*, 1(2), 49-54.

Berndt, J.W., Fisher, J.W., Hevner, A.R., & Studnicki, J. (2001). Healthcare data warehousing and quality assurance. *IEEE Computer,* 34(12), 56-65.

Blaha, M. (2001). *The Managers Guide to Database Technology.* Upper Saddle River, NJ: Prentice Hall.

Blaha, M., & Premerlani, W. (1998). *Object-Oriented Modeling and Design for Database Applications.* Upper Saddle River, NJ: Prentice Hall.

Bockle, G., Hellwagner, H., Lepold, R., Sandweg, G., Schallenberger B., Thudt, R., & Wallstab, S. (1996, June). Structured evaluation of computer systems. *IEEE Computer,* 29(6), 45-48.

Boehm, B. W. (1981). *Software Engineering Economics*. Englewood Cliffs, NJ: Prentice Hall.

Boehm, B. W. (1988). A spiral model of software development. *IEEE Computer,*14(3), 61-72.

Boehm, B. W., Gray, T. E., & Seewaldt, T. (1984). Prototyping versus specifying: A multiproject experiment. *IEEE Transactions on Software Engineering,* SE-10(3), 12-46.

Boloix G., & Robillard P. N. (1995, December). A software system evaluation framework. *IEEE Computer,* 28(12), 17-19.

Booch, G. (1994). *Object-Oriented Analysis and Design with Applications.* Redwood City, CA: Benjamin/Cummings.

Bradley, N. (1998). *The XML Companion.* Upper Saddle River, NJ: Prentice Hall Computer Books.

Bridis, T. (2001, June 27). E-Espionage rekindles Cold-War tensions. *Wall Street Journal, p.* A18.

Brockmeier, J. (2001). *Learn how to speak PHP.* Retrieved March 10, 2003, from the World Wide Web: http://www.linux-mag.com/depts/producreview.html

Brooks, F. P. (1995). *The Mythical Man-Month, Anniversary Edition.* Reading, MA: Addison-Wesley.

Brown, A. L., Ash, D., & Rutherford, M. (1993). Distributed Expertise in the Classroom. *Distributed Cognitions* (pp. 188-227). New York: Cambridge University Press.

Cantor, M. (1998). *Object-oriented project management with UML.* New York, NY: John Wiley & Sons Inc.

Cayne, B., & Lechner, D. (1991). *The New Lexicon Webster's Dictionary of the English Language.* New York, NY: Lexicon Publications, Inc.

Ceponkus, A., & Hoodbhoy, F. (1999). *Applied XML: A Toolkit for Programmers.* New York, NY: John Wiley & Sons.

Chapman, D., & Zwicky, E. (1995). *Internet Security Firewalls.* Sebastopol, CA: O'Reilly & Associates Inc.

Coad, P., & Yourdon, E. (1991). *Object-Oriented Analysis.* Englewood Cliffs, NJ: Yourdon Press/ Prentice Hall.

Cohen, F. (1995). *Protection and security on the information superhighway.* New York, NY: John Wiley & Sons Inc.

Cohen, N. (2001). New Breed of business executives. *Open*, 2(8), 8-11.

Conallen, J. (1999). Modeling Web application architectures with UML. *Communication of the ACM,* 42(10), 63-65.

Connolly, T., Begg, C., & Strachan, A. (1999). *Database Systems: A Practical Approach to Design, Implementation, and Management.* Harlow, England: Addison Wesley.

Cooper, F. J., Goggans, C., Halvey, J. K., Hughes, L., Morgan, L., Siyan, K., Stallings, W., & Stephenson, P. (1995). *Implementing Internet Security.* Indianapolis, IN: New Riders Publishing.

Corrales, J. A. (2001). An asturian view of networking 2015. *Communication of the ACM,* 44(9), 47 – 54.

Correia, E. J., & Finlay, D. (2001). Linux leads as enterprise embraces open-source programs earn respect for cost savings, hardware independence and easy implementation. Retrieved March 10, 2003, from the World Wide Web: http://www.sdtimes.com/news/021/story18.htm#top

Cubranic, D., & Booth, K.S.(2001). *Coordinating open-source software development.* Retrieved March 10, 2003, from the World Wide Web: http://computer.org/proceedings/wetice/0365/03650061abs.htm

Curtis, B., Krasner, H., & Iscoe, N. (1988). A field study of the software design process for large systems, *Communication of the ACM,* 31(11), 1268-1288.

Curtis, P. (1992). Mudding: Social phenomena in text-based virtual realities. *Intertek,* 3(3), 25-33.

Cusumano, M. A., & Selby, R.W. (1997). How Microsoft builds software. *Communication of the ACM,* 40(6), 40-46.

Danesh, A. (1999). *Mastering Linux.* Alameda, CA: Sybex Inc.

Daniel, J., & Zawacki, R. A. (1980). *Motivating and Managing Computer Personnel.* New York, NY: Wiley Interscience Inc.

Davis, P. T., & McGuffin, C. R. (1995). *Wireless Local Area Networks.* New York, NY: McGraw-Hill, Inc.

De, P., & Ferratt, T. (1998). An information system involving competing organization. *Communication of the ACM,* 41(12), 90-98.

Deitel, H. M., & Deitel, P. J. (1999). *JAVA: How to Program.* Upper Saddle River, NJ: Prentice-Hall, Inc.

Dempsey, B. J., Weiss, D., Jones P., & Greenburg, J. (2002). Who is an open-source software developer? *Communication of the ACM,* 45(2), 67-72.

Dibona, C., Ockman, S., & Stone, M. (1999). *Open Sources: Voices from the Open Source Revolution,* Sebastopol, CA: O'Reilly Associates.

Drummond, J.G. (2000). *Open source software and documents: A literature and Online resource review.* Retrieved March 10, 2003, form the World Wide Web: http://www.omar.org/opensource/litreview/index.html#note3

DuBois, P. (2000). *MySQL.* Indianapolis, IN: New Riders Publishing.

Early, P. C. (1986). Supervisors and shop stewards as a source of contextual information in goal setting: A comparison of the United States with England. *Journal of Psychology, 71,* 111-118.

Escamilla, T. (1998). *Intrusion detection.* New York, NY: John Wiley & Sons Inc.

Feiler, J. (1999). *Database driven Web sites.* San Francisco, CA: Morgan Kaufmann Publishers, Inc.

Felton, E.W., & Schneider, M. A. (2000, November 1-4)). Timing attacks on Web privacy. Paper. *7th ACM Conference on Computer and Communication Security.* Athens, Greece: ACM Workshop on Computer and Communication Security.

Fielding, R. T., Whitehead, E. J., Anderson, K. M., Bolcer, G. A., Oreizy, P., & Taylor, R. N. (1998). Web-based development of complex information products. *Communication of the ACM*, 41(8), 84-92.

Finley, D. (2000). *Open-source projects in hands of select few recent survey shatters 'Bazaar' mystique.* Retrieved March 12, 2003, from the World Wide Web: http://www.sdtimes.com/news/008/story1.htm#top

Fosnot, C. (1992) Constructing Constructivism. In T. Duffy & D. Jonassen (Eds.) Constructivism and the Technology of Instruction, A Conversation, pp. 167-176. Hillsdale, N.J: Lawrence Erlbaum Associates.

Fraternali, P. (1999). Tools and approaches for developing data-intensive Web applications: a survey. *ACM Computing Surveys,* 31(3), 227-263.

Friedlein, A. (2001). *Web project management: Delivering successful commercial web sites.* San Francisco, CA: Morgan Kaufmann Publishers Inc.

Galli, P. (2002). *Linux raking in enterprise support.* Retrieved March 12, 2003, from the World Wide Web: http://www.eweek.com/print_article/0,3668,a=22347,00.asp

Garber, L., (2000, October). Linux advances on both desktop and palmtop. *IEEE Computer Society*, 33, 10-13.

Gay, L.R. (1996). *Educational Research.* Upper Saddle River, NJ: Prentice Hall.

Gaur, N. (1999). *Top Open-Source Security Tools For UNIX.* Retrieved March 12, 2003, from the World Wide Web: http://sdnhq.undp.org/lstarch/sdnptech/msg02403.html

Geller, L.N., Alper, J.S., Billings, P.R., Barash, C.I., Beckwith, J., & Natowicz, M. (1996). Individual, family, and societal dimensions of genetic discrimination: A case study & analysis. *Science and Engineering Ethics,* 2(1), 1-5.

Goldberg, A., & Rubin, K. (1995). *Succeeding with Objects: Decision Framework for Projects Management*. Reading, MA: Addison-Wesley.

Goncalves, M., & Brown, S. (2000). *Check Point Firewall-1: Administrative Guide.* New York, NY: McGraw-Hill.

Gorden, L. A., & Loeb, M. P. (2001). Using information security as a response to competitor analysis systems. *Communication of the ACM,* 44(9), 70-75.

Gould, J. D., & Lewis, C. (1985). Designing for usability: Key principles and what designers think. *Communication of the ACM*, 28(3), 300-311.

Goulde, M. (1999, October). Is XML the answer? Depends on the question. *Application Development Trends,* 6, 10-20.

Guterman, J. (2003). *Managing open-source.* Retrieved February 26, 2003, from the World Wide Web: http://www.business2.com/articles/web/print/0,1650,47035,FF.html

Han, D. (2001, August). Why MySQL? *Open*, 2(8), 30-31.

Hasselbring, W. (2000). Programming languages and systems for prototyping concurrent applications. *ACM Computing Surveys*, 32(1), 8-12.

Haviland, W. A. (1974). *Anthropology.* New York, NY: Holt, Rienhart & Winston Inc.

Henderson-Seller, B., & Unhelka, B. (2000). *Open Modeling with UML.* Boston, MA: Addison-Wesley, Inc.

Hildebrand, J.D. (2001) Open source watch: Does open-source still matter? . Retrieved March 12, 2003, from the World Wide Web: http://www.sdtimes.com/cols/opensourcewatch_025.htm

Hill, C.W.L., & Jones, G. R. (1992). *Strategic Management: An Integrated Approach.* Boston, MA: Houghton Mifflin Company.

Hill, J. A., & Misic, M. M. (1996). Why you should establish a connection to the Internet. *TechTrends*. 41(2), 10-18.

Hollandar, J. (2002). *The challenge of the open-source business model.* Retrieved March 12, 2003, from the World Wide Web: http://www.gigalaw.com/articles/hollandar-2000-04-p5.html

Huston, T. (2001). Security issues for implementation of e-medical records. *Communication of the ACM,* 44(9), 89-94.

Huston, T., & Huston, J. (2000). Telemedicine: A practical reality. *Communication of the ACM,* 43(6), 91-95.

Jacobson, I., Christerson, M., Jonsson, P., & Overgaad, G. (1992). *Object-oriented Software Engineering: A Use Case Driven Approach.* Reading, MA: Addison-Wesley, Inc.

*Advantages and disadvantages of each architecture.* JBDC White Paper. (1997). Retrieved March 12, 2003, from the World Wide Web: http://rcs43.urz.tu-dresden.de/~nuehren/jdbc/4_2.htm

Joshi, J.B.D., Walid, W. G., Ghafoor, A., & Spafford, E.H. (2001). Security models for web-based applications. *Communication of the ACM,* 44(2), 38-44.

Johnson, D. J. (1997). Ethics online. *Communication of the ACM,* 40(1), 60-64.

Johnson, R. (2000). The ups and downs of object-oriented systems development. Communication of the ACM, 43(10), 68-73.

Kang, M. H., Park, J.S., & Froscher, J. N. (2001, May 3-4). Access control mechanisms for inter- workflow. Paper. *Sixth ACM Symposium on Access control models and* organizational *technologies*. Chantilly, VA: ACM Workshop on Role Based Access Control.

Kent, S. (1993). Internet privacy enhanced mail. *Communication of the ACM*, 36(8), 48-60.

King, P., & Tester, J. (1999, May). The landscape of persuasive technologies. *Communication of the ACM*, 42(5), 31-44.

Kroenke, D. M., (2002). *Database Processing.* Upper Saddle River, NJ: Prentice Hall, Inc.

Laitinen, M., Fayad, M., & Ward, P. (2000). Thinking Objectively: The problem with scalability. *Communication of the ACM,* 43(9), 105-107.

Laudon, K, C., & Laudon, J. P. (1996). *Managemnet Information Systems.* Upper Saddle River, NJ: Prentice Hall, Inc.

Lea, C., Choi, W., Kent, A., Prasad, G., & Ullman, C. (2000). *Beginning PHP4.* Birmingham, UK: Wrox Press Ltd.

Leavett, N. (2001). Linux: At a turning point. *IEEE Computing 34(9).* Retrieved March 12, 2003, from the World Wide Web: http://www.computer.org/computer/homepage/june/ind_trends/index.htm

Ledgard, H. F. (2001). The Emperor with no clothes: Examining the software problem from a scientific standpoint, *Communication of the ACM,* 44(10), 126-129.

Leedy, L. D., Newby, T. J., & Ertmer, P.A. (1997). *Practical Research: Planning and Design.* Upper Saddle River, NJ: Prentice Hall, Inc.

Lukasik, S. K., Greenberg, L. T., & Goodman, S. E. (1998, June). Protecting an invaluable and ever-widening infrastructure. *Communication of the ACM,* 41(6), 11-13.

Mann, S., & Mitchell, E. L. (2000). *Linux System Security: The Administrator's Guide to Open Source Security Tools*. Upper Saddle River, NJ: Prentice Hall, Inc.

Markus, M. L. (1983). Power, politics, and MIS implementation. *Communication of the ACM,* 26(4), 430-444.

Maruyamaa, H., Tamura, K., & Uramoto, N. (1999). *XML and Java: Developing Web Applications.* Reading, MA: Addison-Wesley Longman, Inc.

Maslow, A. H. (1943). A theory of human motivation. *Psychological Review, 50.* 370-396.

Mayfield, T., Roskos, J.E., Welke, S.T., & Boone, J.M. (1991). *Integrity in automated information systems.* (NCSC Tech. Rep. No. 79-91, pp. 38-48). Alexandria, VA: Institute for Defense Analyses.

McCarthy, J. (1995). *Dynamics of Software Development*. Redmond, WA: Microsoft Press.

McConnell, S. (1996). *Taming Wild Software Rapid Development Schedules.* Redmond, WA: Microsoft Press.

McConnell, S. (1999). Open-source methodology: Ready for prime time. Retrieved March 12, 2003, from the World Wide Web: http://www.construx.com/stevemcc/ieeesoftware/eic06.htm

Medinets, D. (2000). *PHP3 Programming Browser-Based Applications*. New York, NY: McGraw-Hill.

Meeks, B. N. (1997, August). Privacy lost, anytime, anywhere. *Communication of the ACM,* 40(8), 11-13.

Menkus, B. (1988). Understanding the use of passwords. *Computers and Security*, 7(2), 132-136.

Microsoft Corporation (2001). *Bill Gates' letter to Technology Professionals: On why we are building .NET technology.* Retrieved March 12, 2003, from the World Wide Web:http://www.microsoft.com/PressPass/misc/06-18BillGNET.asp

Mintzberg, H. (1973). *The Nature of Managerial Work.* Englewood Cliffs, N.J: Prentice-Hall.

Mockus, A., Fielding, R. T., & Herbsleb, J. (2000). A case study of open source software development: The Apache server. *Proceedings of the 22nd international conference on Software engineering* (pp. 252-261), Limerick, Ireland: ACM Conference on Software Engineering.

Moor, J. H. (1997). Towards a theory of privacy in the information age. *IEEE Computer and Society,*30(9), 27-32.

Moorhead, G., & Griffin, R.W. (1989). *Organizational Behavior. Boston*, MA: Houghton Mifflin Company.

Morris, R. (1985). *A Weakness in the 4.2BSD UNIX TCP/IP Software.* Computing Science Technical Report No. 117, Murray Hill, NJ: AT&T Bell Laboratories.

Muller, P.A. (1997). *Instant UML.* Birmingham, UK: Wrox Press Ltd.

Mynatt, E. D., Adler, A., Ito, M., & O'Day, V. L.(1997). Design for Network communities. *In Proceedings of the ACM Conference on Human Factors in Computing Systems* (pp. 210-217), New York, NY: ACM Press.

Myers, B., Hollan, J., Cruz, I., Bryson, S., Bulterman, D., Catarci, T., Citrin, W., Glinert, E., Gruden, J., & Ioannidis, Y. (1996). Strategic directions in human-computer interface. *ACM Computing Survey,* 28(4), 791-818.

MySQL (2002). *State of Rhode Island Saves Development Time and Costs with MySQL and Open Source.* Retrieved February 25, 2003 from the World Wide Web: http://www.mysql.com/press/user_stories/srisdt.html

MySQL (2003). *Westone amplifies database performance with MySQL.* Retrieved February 25, 2003 from the World Wide Web: http://www.mysql.com/press/user_stories/westone.html

Nemeth, E., Snyder, G., Seebass S., & Hein, T.R. (1995). *UNIX Systems Administration Handbook.* Englewood Cliffs, NJ: Prentice Hall.

Neumann, P. G. (1993). Risk considered globally. *Communications of the ACM*, 36(1), 154-156.

Neumann, P. G. (1999). Inside risks: robust open-source software. *Communications of the ACM*, 42(2), 128-130.

Open Source Initiative (2001). *The Open-source case for business.* Retrieved March 12, 2003, from the World Wide Web: http://www.opensource.org/advocacy/case_for_business.html

Oppliger, R. (1997). Internet security: Firewalls. *Communication of the ACM,* 40(5), 92-102.

O'Reilly, T. (2000). Open-source: The model for collaboration in the age of the Internet. Retrieved March 12, 2003, from the World Wide Web: http://www.oreillynet.com/pub/a/network/2000/04/13/CFPkeynote.html

Park, C. S. (1999). *XML -The foundation for the future.* Retrieved March 12, 2003, from the World Wide Web: http://wcb.ucr.edu/ wcb/ schools/AGSM/mgt/jgerdes/3/forums/forum2/messages/169.html

Parnas, D.L., Schouwen, A.J., & Kwan, S.P. (1990). Evaluation of safety-critical software. *Communication of the ACM,* 33(6), 636-648.

Peden, M., & Young, G. (2001). From voice-band modems to DSL technologies. *International Journal of Network management,* 11(6), 265-270.

Perens, B. (2002). *The open-source definition.* Retrieved March 12, 2003, from the World Wide Web: http://www.perens.com/OSD.html

Perlow, J. (2001). *Caldera's quick start to Linux.* Retrieved March 12, 2003, from the World Wide Web: http: www.linux-mag.com/depts/productreview.html

Phillips, P.M., & Spakovsky, H. A. V. (2001), Gauging the risk of Internet elections. *Communication of the ACM,* 44(1), 73-80.

Poole, J., Barkley, J., Brady, K., Cincotta, A., & Salamon, W. (1999). *Distributed Communication Methods and Role-based Access Control for use in Health Care Applications.* Technical Report. National Institute of Standards and Technology. (NISTIR 5820).

Porter, M. E. (1985). *Competitive Advantage: Creating and Sustaining Superior Performance.* New York, NY: Free Press.

Qusterhout, J. (1999). Free software needs profit. *Communication of the ACM,* 42(4), 44-45.

Railsback, K. (2001). *IBM suit answers call.* Retrieved March 14, 2003, from the World Wide Web: http: www.linux-mag.com/depts/productreview.html

Raymond ,E. S. (1999). *Open Source: Programming as if Quality Mattered.* Retrieved March 14, 2003, from the World Wide Web: http://speakout.com/activism/opinions/3122-1.html

Raymond, E. S. (2000). *The jargon file.* Retrieved March 14, 2003, from the World Wide Web: http://www.tuxedo.org/~esr/jargon/

Raymond, E. S. (2001). *The Cathedral & the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary.* Sebastopol, CA: O'Reilly.

Reason, J. (1990). *Human Error.* Cambridge, UK: Cambridge University Press.

Retting, M. (1990). Software teams. *Communication of the ACM*, 33(10), 23-27.

Ricoeur, P. (1974). Nature and freedom. *Political and Social Essays*, Athens, OH: The Ohio University Press.

Rieken, B., & Weiman, L. (1992). *Adventures in Unix Network Applications Programming*. New York: John Wiley & Sons, Inc.

Riel, M., & Levin, J. (1990). Building electronic communities: Success and failure in computer networking, *Instructional Science,* 19(1), 145-169.

Rindfleisch, T. C. (1997). Privacy, information technology, and healthcare. *Communication of the ACM,* 40(8), 92-100.

Rubinstein, D. (2002). *Feather in Apache's cap*. Retrieved March 14, 2003, from the World Wide Web: http://www.sdtimes.com/news/054/story3.htm

Schach, S. R. (1998). *Classical and Object-Oriented Software Engineering*. New York, NY: McGraw-Hill.

Shankland, S. (1999). *Apple opens part of its OS.* Retrieved March 14, 20032, from the World Wide Web: httpd://news.cnet.com/news/0-1003-200-339995.html?tag

Shankland, S. (2001). *Apple hires open-source leader OS.* Retrieved March 14, 2003, from the World Wide Web: httpd://news.cnet.com/news/0-1003-200-684175.html

Schneier, B. (2000). *Secrets and Lies*. New York: John Wiley & Sons, Inc.

Siau, K., Lim, E., & Shen, Z. (2001). Mobile commerce: Promises challenges and research agenda. *Database Manager*, 12(3), 4-17.

Siau, K., Nah, F. F., & Teng, L. (2002). Acceptable Internet use policy. *Communication of ACM*, 45(1), 75-79.

Silberschatz, A., Korth, H. F., & Sudarshan, S. (2002). *Database systems concepts*. New York, NY: McGraw-Hill Companies, Inc.

Simmers, C. A. (2002). Aligning Internet usage with business priorities. *Communication of the ACM*, 45(1), 71-74.

Sonnenreich W., & Yates, T. (2000). *Building Linux and OpenBSD firewalls*. New York, NY: John Wiley & Sons, Inc.

Spafford, E. (1989). The Internet worm: Crisis and aftermath. *Communication of the ACM,* 32(6), 678-688.

Spiegel, R. (1999, October 20). Yankee group foresees 10 million networked digital homes by 2003. *E-Commerce Times.* Retrieved March 14, 2003, from the World Wide Web: http://www.ecommercetimes.com/news/articles/991020-3.shtml

Steiner, G.A., & Steiner, J.F. (1988). *Business, Government, and Society.* New York, NY: Random House Inc.

Sturdivant, F. D., & Ginter, J. L. (1977). Corporate social responsibility: Management attitudes and economic performance. *California Management Review.* (??)

Sutor, R. H. (1997). *An educational executive information system for public school district superintendents.* Unpublished Doctoral Dissertation Proposal. Nova Southeastern University, Graduate School of Computer and Information Sciences, Fort Lauderdale. Fort Lauderdale, FL.

Swinyard, W. R., Rinne, H., & Keng Kau, A. (1990). The morality of software piracy: A cross-cultural analysis. *Journal of Business Ethics,* 9, 655-662.

Taylor, G., & Shim, J. P. (1993). A comparative examination of attitudes toward software piracy among business professors and executives. *Human Relations,* 46, 420-431,

Taylor, L. (2002). *Client/Server Frequently Asked Questions.* Retrieved March 14, 2003, from the World Wide Web: http://www.faqs.org/faqs/client-server-faq/

Thompson, A.A., & Strickland, A.J. (1984). *Strategic Management: Concepts and cases.* Plato, TX: Business Publications Inc.

Tseng, S., & Fogg, B.J. (1999, May). Credibility and computing technology. *Communication of the ACM,* 42(5), 39-43.

Urban, M., & Tiemann, B. (2002). *FreeBSD Unleashed.* Indianapolis IN: Sams Technical Publishing.

Vacca, J. (1998). *Intranet Security.* Rockland, MA: Charles River Media Inc.

Verner, J.M., & Cerpa, N. (1997). The effect of department size on developer attitudes to prototyping. *Proceedings of the 1997 international conference on Software engineering* (pp. 445-455), Boston, MA: ACM Conference on Software Engineering.

Wahba, M. A., & Bridwell, L.G. (1976). Maslow reconsidered: A review of research on the hierarchy theory. *Organizational Behavior and Human Performance,* 212-240.

Walmsley, P. (2001). *Definitive XML Schema.* Upper Saddle River, NJ: Prentice-Hall, Inc.

Weaver, A.C., Vetter, J.R., Whinston, A.B., & Swigger, K. (2000, October). The future of E-Commerce. *IEEE Computer Society,* 33(10), 33-40.

Weber, J. (1993). Exploring the relationship between personal values and moral reasoning. *Human Relations*, 46, 434-364.

Weinstein, L., & Neumann, P. G. (2000). Internet Risks. *Communication of the ACM*, (43)5, 144–145.

Westfall, R. (2001). Hello, world considered harmful: It all starts at the beginning: OO programming learned naturally, not procedurally. *Communication of the ACM*, 44(10), 219-141.

White, G. B., Fisch, E. A., & Pooch, U.W. (1996). *Computer System and Network Security*. Boca Raton, FL: CRC Press.

Zari, M., Saiedian, H., & Naeem, M. (2001). Understanding and reducing Web delays. *IEEE Computer*, 34(12), 30-37.

Zucconi, L., Mack, G., & Williams, L. G. (1990). Using object-oriented development for support prototyping. *Proceeding of the 12th international conference on Software engineering (pp. 129-132)* Nice, France: ACM Conference on Software Engineering.