To the Graduate School:

The members of the Committee approve the thesis of Seaghan UiBreaslain presented on July 29, 2003.

Rex

Rex Gantenbein

James Ward

Jim Ward

Jerry Ham

Jerry Hamann

APPROVED:

Jeffrey Van Baal

Jeffrey Van Baalen, Head, Computer Science Department

Don Roth

Don Roth, Dean, The Graduate School

UiBreaslain, Seaghan, Software Engineering and Free Software: a Small Evaluation, M.S., Department of Computer Science, August, 2003.

In this study, I compare a Linux-based, free software, engineered Web-accessible database system to a similar Microsoft Windows-based, nonengineered system. Many studies are done to rate the speed or efficiency of a variety of software and hardware. While this project was too small to provide any new information on the power of the software used, it does give some insight into the value of software engineering, even in very small projects.

The free and proprietary software turned out to be roughly equivalent, in terms of power, capabilities, and usability, for my purposes. Where the two projects differ is not in what software was used to build the systems, but in the way the systems were designed and implemented.

The engineered system had most of the work done in the planning phase. Writing the code then proceeded quickly, and a long and leisurely testing phase followed. The non-engineered system was constantly behind schedule, periodically redesigned on the client's whim, and tested "on the fly": problems were fixed in the next rewrite.

SOFTWARE ENGINEERING AND FREE SOFTWARE: A SMALL EVALUATION

by

Seaghan UiBreaslain

A thesis submitted to the Department of Computer Science and The Graduate School of the University of Wyoming in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE
in
COMPUTER SCIENCE

Laramie, Wyoming
August, 2003

“Proper Previous Planning Prevents Piss-Poor Performance”

Navy UDT Chief Petty Officer

Everett E. Barrett’s Law of the Seven P’s

as quoted in Dick Marcinko and John Weisman,

Rogue Warrior: Task Force Blue

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The purpose of this project was to create, with free software, a working Web-accessible database and online ordering system, and compare various aspects of this system to a similar system built with Microsoft (or other proprietary) software. Also, I compare the software engineering approach to a more organic, unplanned approach. Usually, software engineering is used on large, “official” projects, and small projects just "happen.” However, that did not turn out to be the case with this study. Originally, I had intended to compare free, open source software and proprietary software used to build somewhat similar websites, one under controlled conditions, the other under real life working conditions, but both using software engineering methods.

Real life, as it has a way of doing, changed that.

The free-software project was constructed for the Anime Club, Laramie Chapter (ACLC), a student club that provides an opportunity for UW students to view Japanese animated television programs and movies, and to learn something of the Japanese culture. I was elected Vice President of the ACLC in Spring of 2002. The VP's duties mostly involve the care and maintenance of the club's videotape library. I saw an opportunity to practice software engineering, and "try out some cool new software," while making my duties as Vice President of the ACLC easier. I would design and build an online database to keep track of the library, with tables and queries to allow club members to search for works by their favorite actors or studios, and an online order form, to email requests from members to allow better tracking of tape check-outs. I planned to do this in as controlled a manner as possible, because I could. Since I was client, designer, and programmer, I could model the whole software engineering process.

I was hired in September, 2002, by the Wyoming State Planning Grant to build a public information website, that would eventually include Web-accessible databases. I would be working for a client, with Microsoft products that the client provided. I saw that this project had the potential to be a "real world" version of the laboratory exercise I was doing for the ACLC. I hoped to be able to compare proprietary Microsoft products used on a production website to the free software I was using on the ACLC Website, to see if the free equivalents really were equivalent. I met with members of the SPG to design the website, and plan the release schedule. When the first release was presented, the plan collapsed. The original schedule was abandoned, and priorities were rearranged, often on a daily basis. As the websites became more alike in size and function, the approach to building them became more different.

Chapter 2

Overview of the Projects

2.1 Anime Club Laramie Chapter

Founded by Darren Bills, Joe Justice and Rob Earhart in 1998,

> "The Anime Club Laramie Chapter (ACLC) is a University of Wyoming Recognized Student Organization dedicated to increasing the awareness and popularity of Japanese animation, or anime (pronounced: An-EE-May) to the residents of Laramie, Wyoming. To do this, we provide regularly scheduled, free, entertaining, and enlightening experiences through public showings of Japanese Animation. Our organization exists to provide a forum through

> which participants will be both entertained and educated. This is because Anime provides an excellent window that Americans can use to begin learning about the culture of Japan. In addition, the Laramie Chapter gives it members the chance to make new friends and acquaintances. There will also be occasional contests to win free anime and J-Pop related stuff, like tapes, soundtracks, and clothing." [3]

The ACLC maintains a library of anime tapes and DVDs for members to check out to watch at home. This library is the main responsibility of the Vice President of the club. Previously the library consisted of several cardboard boxes of tapes, stored wherever they could be stored. My predecessor kept them in a pile in the middle of his bedroom floor. When I received the library, the cardboard boxes had been replaced with plastic storage bins. The only record of the contents of the library was kept by the President, not the Vice President. Checkout tracking was done by having the member write their name, tape checked out and date checked out, on a sheet of paper. When the tape was returned, the Vice President drew a line through the entry. Orders were placed by emailing the Vice President, or requesting tapes in person. The VP would usually write the order on his hand in ballpoint pen. Only three tapes were lost, and it is not clear when they were lost, as no inventory verification had been done when the library was passed from the outgoing VP to his replacement.

I had used the idea of an online database and ordering system for the ACLC library as the basis for a semester project in Database Systems class. Then when I was elected Vice President of the ACLC, I decided to implement the database for my own use. I hoped that someone would be interested in implementing and especially in hosting the online ordering system, but since most of the ACLC members are not computer science students, I only planned on doing what I could in my "copious free time" [4] and did not expect any assistance. Then, when I needed a thesis project, I was able to combine the database with the ordering system, as an evaluation of free software for Web use.

2.2 State Planning Grant

> "The Wyoming Department of Health (WDH) received the State Planning Grant in July 2002 from the Health Resources & Services Administration (HRSA). These competitive twelve-month

> grants are designed for states to study their uninsured population and to develop options to provide access to high quality, affordable health coverage for all State citizens." [5]

The Department of Health set up the State Planning Grant Project, which then contracted with the University of Wyoming to study the population of Wyoming and the nature and distribution of health insurance, and form a plan to provide health insurance coverage to the currently uninsured people of the State. The State Planning Grant wanted to have a public information Web site where they could list the members of the task-force, announce public events, post links to information resources, and get feedback from the public.

I was hired as a graduate research assistant to build the public information website for the State Planning Grant. As this agency is funded by a grant from the federal government, they do not have the same monetary constraints as the ACLC. Also, they can share hardware and software with the Center for Rural Health Research and Education (CRHRE).

2.3 Software Options

The ACLC and SPG websites have many similarities: static HTML text pages, dynamic HTML database search and modification pages, the underlying database programs and scripting languages. The design philosophies differ however, and this affects the choice of particular software packages.

Due to limited financial resources, my own as well as the Anime Club's, cheap or free programs are the best choice for the ACLC project. The University has some software available for students to use on the *ASUWlink.uwyo.edu* webserver, but this does not include any server-side script processors or databases, nor do they plan to add them[1]. Therefore, some other host machine with the necessary software had to be found.

The SPG has significantly greater resources and can share software and hardware with CRHRE. The Center for Rural Health Research and Education has a number of servers, and site-licenses for Microsoft programs, such as Windows.

[1] The UW IT Department "might" consider installing such programs for research purposes, at the instructor's written request, but would not let ordinary student organizations such as the ACLC use them after the active research phase ended, to preserve bandwidth and computing cycles for academic use and to reduce the chance of viruses/"hacking." [6]

In the next section I discuss some of the options available, how they might apply to each project, and what was actually chosen. Some of these choices were predetermined. The software must be free for the ACLC since it had no software available and no money to spend on it. In the case of the SPG, it was better to use software that was available in Microsoft packages already owned by CRHRE, such as ASP and Access, so more of the grant money could be spent on the actual research.

Even within a category, open source database servers for instance, there are choices to make: both POSTgreSQL and MySQL are included in Red Hat 7.3, so are there overwhelming reasons to use one or the other?

2.3.1 Open Source

First, since there can be some confusion on the subject of "free software" and "piracy" this subject bears further discussion.

Open source software is often, but not necessarily, free, i.e. zero price. Often there is at least a charge for distribution and media. There may be license restrictions on its use, as well. All that is required to be open source is that the source code is not kept secret. "[O]pen source software is software whose source code is freely available." [7]

> "The basic idea behind open source is very simple: When programmers can read, redistribute, and modify the source code for a piece of software, the software evolves. People improve it, people adapt it, people fix bugs. And this can happen at a speed that, if one is used to the slow pace of conventional software development, seems astonishing." [8]

Richard Stallman, founder of the free software movement, makes the distinction between "freeware" - programs that are given away only as binaries without the source code, "free software" - programs that are not and can never be made proprietary, nor can any program based on them (the "GNU virus")[2], and "open source" - programs for which source code is available,

[2] "...The G[NU] P[ublic] L[icense] does not permit incorporating a GNU program into proprietary programs. This ... requirement leads to what is jokingly referred to as the "GNU virus," since use of any GNU software in proprietary commercial software can lead to the software being covered by the terms of the GNU GPL. This provision has limited the amount of GNU software incorporated into commercial products..." [9] In other words, any program covered by the GPL is "free," and any program incorporating code from a program covered by the GPL is itself then covered by the GPL, thus "free." If Microsoft used GPLed code in Windows, Windows in its entirety would then be "free" software. The source code would have to be made available to the general public.

but for which proprietary (binary distribution only) modifications are allowed. He says free software should be considered in terms of "free speech, not free beer." [10].

However, there may be other costs involved, such as buggy programs, awkward interfaces, lack of GUI, poor if any documentation, etc. Sometimes a great program will simply be abandoned because the original developer has no time or money to continue, or loses interest, and no one else wants to take over. Not every zero price product is free. Sometimes, as the song says, "it costs you nothing , and it's worth the price."

With all the hassles involved in "free software," unreliable support, poor documentation, the "GNU virus" etc., why bother? There are many proprietary software packages available that perform the same functions, possibly better. Other than "getting something for nothing" or "getting back at Micro$oft" are there advantages to "free software?" To build the ACLC Library project, I needed an HTML editor, database server, a scripting language to handle access to the database, an http server, and a host computer and operating system. Obviously there are proprietary products available that will perform all of these duties. What are the disadvantages of the proprietary software that make the free alternatives so much more attractive, other than cost?

Not all free software is worthless. The Linux operating system itself is an example of this. While some people will use anything, no matter how bad, as long as it is not made by Microsoft, some users such as ISPs, can not afford to use free software that is no good. Linux has made a name for itself as a webserver OS by having features and reliability[3].

Linux is a free Unix-like operating system based on the GNU project founded by Richard Stallman [10], and the Linux kernel, originally written by Linus Torvalds [14]. The software packaged in a Linux "distro" is mostly free or open source. Some distributors may add their own proprietary programs, but most do not. I used Red Hat 7.3 for the ACLC project, mainly because it was the latest release at the time I requested a server for my website, so it is what was installed. It is also the version I use on my personal

[3] The advertisers of Linux Journal magazine can be expected to be enthusiastic about the operating system. The advertisers in Network World and eWeek do not have the guaranteed fan-base of the dedicated Linux publications. So the market must really exist for IBM to run multiple full page adds, including a two-page spread inside the cover Network World [11], for their Linux systems. Also witness the recent collaboration between Dell and Red Hat [12]. Even Sun now offers support for various flavors of Linux as well as its own Solaris Unix OS on its servers. [13]

computers. It includes the Apache webserver, MySQL, PHP, and the KDE window manager.

The KDE window manager provides a desktop environment similar to Windows 98/2000. The KDE package comes with an office suite, (basically a repackaging of Corel Office/WordPerfect 8)[4] and Konqueror, a Web and file browser much like Microsoft Windows Explorer/Internet Explorer, but with fewer bugs. The user can switch between up to 16 desktops, each with an assortment of programs displayed in its own window. Each desktop is like the Microsoft Windows desktop in that windows containing various programs and files can be moved around, stacked or shuffled. Windows can be moved from one desktop to another, unlike the single-desktop environment of Microsoft Windows. Also, command line terminals can be opened in their own windows, each as a different user login, if necessary. The Gnome windows manager is also included, and is more like Macintosh than Windows. All programs are available through either environment.

The main differences between Windows 2000/XP and an up to date

[4] Corel purchased WordPerfect several years ago and also produces a Linux distribution. WordPerfect was ported to Linux, and the KDE office suite looks very similar to the old Corel Office 8/WordPerfect 8 suite. It is not completely compatible with recent Microsoft products [15].

Linux/KDE install are the cost, "look and feel" and extra programs. Linux is available as a free download, or can usually be purchased as a set of CD-ROMS and some paper documentation for significantly less than $100. The distro includes KDE and other windows managers, Apache, PHP, MySQL, POSTgreSQL, several text editors, full-scale word processors and other office programs, an assortment of network utilities such as telnet and ftp clients and servers, and many other things, useful and otherwise. Much of this software is tricky to install, or at least has a "scary" command-line interface. Almost any problem can be fixed, but it can take days to find the file that needs editing to repair it. I chose to work with Red Hat Linux for the ACLC project because 1) it is available on the computers in the Computer Science Department, and I am already familiar with the environment, 2) my own personal computers run the same version, so I could work on my home computer, without worrying about code portability, 3) it had all the programs I needed in a standard install, though some needed to be activated, 4) it's free.

MySQL is a relational database management system. For Internet applications, it is faster than POSTgreSQL [2]. It is available under both open source and commercial licenses (for those applications that should not be exposed to the GNU virus) [16]. Some of the features I originally designed for the ACLC database are not available in MySQL, such as new member-ID numbers assigned by triggers, but the "work-around" using MySQL turned out to be a cleaner design.

The advantages of MySQL for the ACLC are the price (free) and the integration with the other free program, PHP. MySQL is fast and powerful enough to use for websites getting thousands of hits per day, which is overkill for the ACLC. Like all the database programs, it has to be installed on a server, which I am providing through the duration of this study. At the end of my research the information in the database can be extracted and placed in another database on some other machine, if we can find one. My hope has been that the database would prove useful enough that someone in the club would want to maintain it. It now appears that the incoming President of the ACLC will provide server space and programs for the club.

PHP is a server-side scripting language designed for the Web [16] PHP code can be embedded in HTML documents. Like ASP and Perl/CGI scripts, PHP scripts are processed by a program running on the server. Among other things PHP can be used to connect to databases, to read or modify the contents of the database and dynamically generate a customized Web page to suit the individual needs of the user. Since it is run on the server, there is no need for the user to have a "PHP-enabled" browser. Even old text-only browsers like Lynx can handle PHP (or ASP), because they actually never see it. Before the page is served, the script is processed, and the result is written as plain HTML. However, if the output is displayed as a table, as in database search results, text-based browsers will not display properly

Why use PHP and not Java, JavaScript, ASP, etc, to connect to the database? PHP is available as a free download for Windows or Linux. It is also included in current Red Hat Linux distributions. However, the same is true for Java. Java is a general purpose language with a reputation for slowness in Internet applications (there is even an alleged internal Sun memo advising Sun employees not to use Java in Sun's own production software circulating on the Internet [17]), while PHP is specifically designed for online database access (as well as general purpose scripting use) and is often packaged with MySQL as an integrated system for that purpose. Further, Java is dependant on client-side support which may not be installed on a user's computer. Java would avoid the necessity of installing a program on the server, but not everyone would be able to use it.

JavaScript is a client-side scripting language supported to some degree by all current browsers (version 4 and higher). As such, it does not need to be installed on the server, reducing download time and server processing. However, many users turn off the JavaScript processing of their browser specifically to prevent client-side applications from running, in case these are hostile (viruses or cracking programs), or just to speed up web surfing.

Further, JavaScript does not have built-in database access functions. They can be built by the user, or the extensions purchased from Netscape. In the case of the extensions, they are not appropriate for the ACLC project because they are not free. The extensions are also run on the server, which wipes out the client-side advantage of regular JavaScript, although it does get around the problem of users deactivating JavaScript processing in their browser.

2.3.2 Microsoft

Microsoft Windows is the most common desktop operating system today because of a number of factors. The convenience of a windowing GUI is undeniable, and the many coordinated Microsoft products offer a mostly consistent workspace, with drag-and-drop transfer of files from one program to another.

For serving HTML pages, Windows has its own IIS Web server with ASP, or Apache has been ported to Windows, due to its success in the Unix/Linux world. PHP and MySQL are both available for Windows, as are Java, SQL Server, Access, and a number of other scripting languages and database servers.

Microsoft Windows is "user friendly" to a fault. Almost nothing uses a command-line interface, and much of the administration is automated. With the server version you get the IIS Web server with ASP. Word and FrontPage are available, at extra cost, as part of the Microsoft Office suite [18]. The standard fix for many serious problems is to reinstall Windows, the problem program, or both.

A good, simple text editor for Windows is hard to find. Notepad is extremely basic, lacking a search/replace function and line numbering. Wordpad has these features but adds some formating that can confuse browsers and cause problems with Web pages. Word and FrontPage cost extra and have (too) many automated extras (talking paper-clips and auto-formatting, among many others) that make life difficult for people who prefer plain, old-fashioned, non-automated text editors such as vi and emacs. For the SPG project, since it was Microsoft-based, I had to find a compromise editor that would not add formating, tags and style sheets that I did not want, but which could highlight tags and keywords, display line numbers, and perform search-and-replace operations.

Microsoft FrontPage, and other programs like it, such as Netscape Composer and Microsoft Word, allow the user to create web-pages without knowing HTML. The user may write text and insert graphics without bothering with the technical details of how it is done. However there are two drawbacks to this approach.

First, part of the reason for doing this project is to learn HTML, PHP, and as much of the technical inner workings of web-pages as possible. Working in a WYSIWYG environment "protects" the user from such inner workings.

Second, these automatic code-writing programs produce much more code than is strictly necessary to generate the page. Netscape Composer, for instance, puts `<font>` tags around each line of a block of text, rather than formating it as a block. FrontPage also uses proprietary extensions to standard HTML, and predefined "web-bots" to perform various functions. Both of these "features" make debugging difficult if not using FrontPage.

Using a basic text editor such as Windows Notepad, vi or kwrite allows rewriting the page on any computer with a basic text editor. Kwrite is much like Notepad in that it produces minimally formatted text, but it has many of the features found in Wordpad such as search/replace. Kwrite is included in the KDE window manager software suite for Linux. Most of the MySQL table-creation and population scripts for the ACLC project were written in kwrite, the rest in vi.

Basic text editors may also be accessed easily over a telnet connection through a phone line and modem. The file size is smaller, and reduces wait-time on slow lines.

Also, pages written with FrontPage extensions require those extensions be installed on the server hosting the page in order to run properly. This reduces the portability of the code. Since the scripting languages and database also need to be installed on the server, this is not as big a difference as it could be. However, if the owner of the server has not installed FrontPage extensions and does not wish to do so, pages written using those extensions will not function fully.

I eventually settled on 1stPage 2000, which is still available for download at freewebmasterhelp.com [19], though Evrsoft, the producer, no longer offers it at their site [20]. This program has all the features I needed and none of the ones that I wanted to avoid. It also has features to "tidy" HTML, finding unclosed tags and other problems, and to convert HTML to XML. I did not use these particular features on this project, but I will use them in the future.

The Access database program is part of the MS Office suite of programs. It provides a reasonably powerful database with a relatively simple GUI for small office or home office use. But, Access is a database, not a database server. It is intended to provide a single database on a single computer. Thus it does not have many of the features and capacity that are necessary to provide database access over the Internet. It will work for small databases with very limited use, and either the SPG or ACLC project could use Access, but even these small projects should be implemented with a program designed as a database server [21].

SQL Server is a database server from Microsoft. The major drawback of it for the ACLC project is that it is not free, in either the "free beer" or "free speech" sense. A quick survey of online shopping sites reveals that SQL Server starts at about $1200. It would have to be purchased and then installed on the IT Department's servers. SPG has access to copies licensed to CRHRE, and its own servers, so price and installation are not issues.

ASP is an extension to a web-server to allow server-side script processing [21]. It is included as part of Windows server versions starting with IIS

4.0 in NT 4.0 [21] through IIS 5.0 in Windows 2000 Server [22] to Windows Server 2003 with IIS 6.0 [23]. Thus it is equivalent to PHP. It has been ported to Apache, but the Apache version lacks JavaScript or VBScript support so the scripts must be written in Perl or the commercial version purchased [24] [25] [26]. I am using ASP with the SPG project, and while using it with the ACLC page would give me an opportunity to work with Perl, PHP is already included in the Red Hat 7.3 distribution I am using. ASP for Unix would have to be downloaded and installed, which goes against the "run what you brung" spirit of the ACLC project, (a philosophy popular in auto racing, in which contestants can not rely on backup vehicles or a trailer full of replacement parts but have to make do with one vehicle, as-is).

The SPG did not have the question of which software to use as they are sharing hardware and software with CRHRE. The software is from Microsoft for the most part, though other companies' products are also included: Adobe PhotoShop and Acrobat, Sawmill traffic analysis program, and 1stPage 2000. We decided to avoid use of some programs such as FrontPage and Word, and design elements such as style sheets, for programming and interoperability reasons. The excess code generated by FrontPage made adapting some preexisting pages difficult, and it was easier and faster to rewrite them from scratch than to convert the old one. We also found that while style sheets are intended to make website display more consistent and reliable over a range of browsers, the browsers that we could expect current users to be running have such poor and inconsistent support for style sheets that we abandoned them in favor of tables.

The W3C recommends against tables for formatting as text-only browsers and site readers for the blind do not handle them well. I took this into account and tried to design the pages to "fail gracefully," i.e. have a readable backup for scripts and use tables only to position cell content, not to order it. Non-JavaScript-enabled browsers are provided with text-only links, pictures all have `<alt=''image - description''>` tags, and the tabular structure of the pages collapses into a simple sequence of links followed by content. Only the database search results do not fail gracefully. With more time the table structure could be redesigned for better text-only display.

XML is outside the scope of this study. This study compares software engineering with non-engineered design, and PHP/MySQL with ASP/Access. While either and both of these websites might have benefited from the use of XML, it would have added an extra complicating factor to the comparison. Word and FrontPage use XML, as well as style sheets, which is part of the cause of the excess code generated by these programs.

2.4 Size and Scope of the Projects

2.4.1 ACLC

There are approximately 20 - 30 active members in ACLC. These are only the registered, dues paying members who may check tapes out of the library. Several more people regularly come to the meetings who do not pay dues. Many others use the website who do not live in Laramie. The ACLC site contains several in-depth reviews of obscure anime programs for which it is hard to find information and is linked from the Anime Turnpike website as a source of information on these anime [27] [28].

The ACLC tape library consists of around 200 VHS tapes and an increasing number of DVDs. These videos contain movies and/or episodes of television programs. The club purchases new videos, members donate videos, and several anime distributors provide promotional copies of new releases. The club needs a database of all videos that is search-able by language, format, medium, as well as title. New videos will be added, and some older ones will need to be removed as their licensing status changes.

Last year (2001-2002) several tapes were lost because the only tracking method was a sheet of paper on which members wrote their name, tape title, and date checked out. When a tape was returned, the member wrote the date returned and drew a line through the entry. When the sheet filled up the VP got a new one and the old one went away. No long term tracking was provided, so which tapes were checked out to whom was unknown at the end of the year. For 2002-2003 the checkout sheets are a printed form. Old sheets are kept in a folder for future reference. Eventually, the database will track which tape is checked out to which member, date out, date in, etc.

Anime fans often know the production details of the anime, who wrote or directed it, which studio produced it, etc., but would like to know what language a particular tape is in. Many people have a preference for or against subtitles, for instance. To the hardcore fan, subtitles are usually preferable to English dubs that may change the story, and that are often very poorly acted. However, subtitles may be too threatening for new fans who want to explore the art and storytelling of anime without having to suffer through a "foreign film." Therefore language will be one of the main search criteria.

According to the hit-counter, the ACLC website has had over 69,500 hits between June 1, 2001, and June 23, 2003. This averages to 643.5 hits per week. Peak traffic in the last year (6/2002 - 6/2003) occurred in July [29]. We can expect traffic to be under 1000 hits per week The number of searches should stay below 100 per week. This is approximately the same amount of traffic as the SPG public site is expected to receive.

2.4.2 SPG

The SPG site pages consist of static textual content inside a templated framework. This framework (not frames) contains the SPG logo, in-site links, and pictures. The structure is provided by tables, as mentioned earlier. A remote script provides consistent links on all pages and easier maintenance. Three dynamic HTML pages, the contact page, database search page and database modification page, use ASP scripts to generate their code. The script is inserted in the content section of the standard template.

The SPG site is new and does not have the established user base and outside linkage that the ACLC has. Therefore traffic is currently lower: 949 hits between November 5, 2002, when the counter went online, to July 3, 2003. Many of these are internal to the SPG: I and others in the organization had to view the site to check display and speed. The intended user-base of the site is potentially the entire population of Wyoming, as it becomes better known throughout the State.

Chapter 3

Implementation

3.1 ACLC

The ACLC project is running on a server in the Computer Science Department. Server *wks16.cs.uwyo.edu* is a dual 400MHz Pentium II machine with 125844 KB RAM. The relevant software includes Red Hat Linux 7.3, Apache web server, MySQL v.3.23.54, and PHP 4.1.2. The speed of the computer, based on "feel," or my perceived wait time, is roughly equivalent to the 800 MHZ Pentium III workstations in the student computer labs.

I used a spiral plan for this project: design, build, test, get feedback, fix, design and build next installment, test, etc.... This technique is summarized

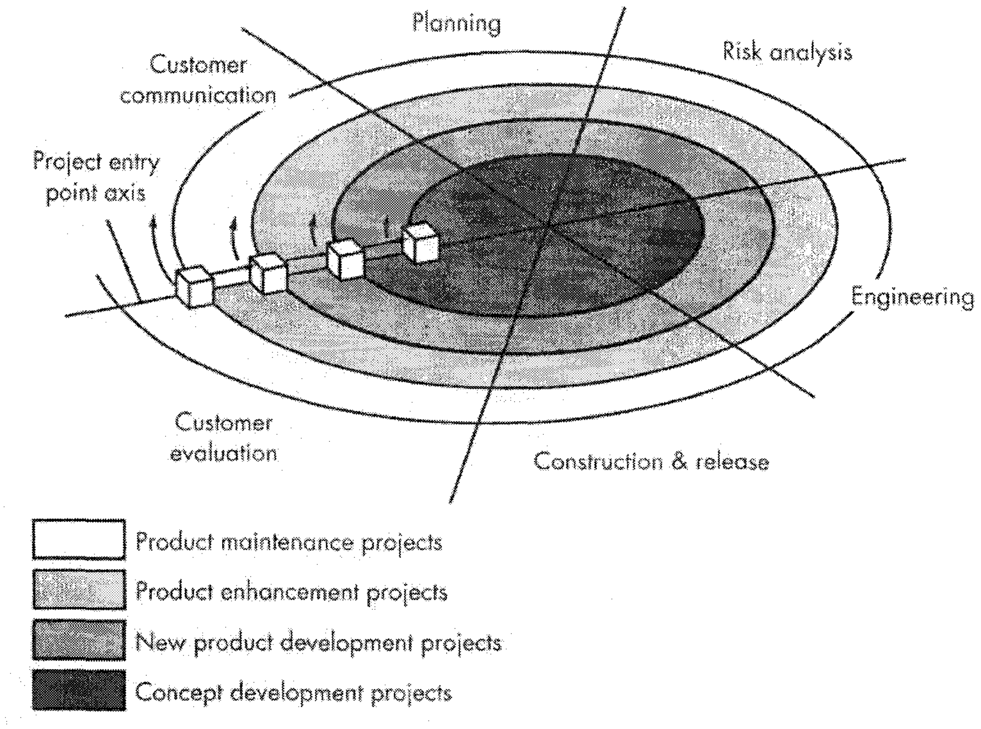

Figure 3.1: Spiral Plan, from Pressman [1]

in Figure 3.1.

3.1.1 Database

The first thing to be built was the member and password tables to authenticate users. Before anyone can be screened out of the order page, we need a list of who belongs in it. Next came the library database. Until it was built, members could view a static list of all the tapes in the library, or at least those that had been entered so far. The library database consists of three tables at this point: The ANIME table listing all titles, the TAPE table listing physical VHS tapes and DVDs, and the TAPE_ANIME table referencing which movies/episodes are to be found on which tape. This is the stage the project is in at this time.

The next phase is to add additional tables giving more detail and scripting to allow more complex searches. Tables containing information on directors, actors, studios etc. will be added, along with the relationship tables to link them.

Debugging of the database was fairly fast and straightforward as the design for most of the initial tables had been worked out and tested in Database Systems class, and new tables were based on the same pattern. Most debugging was needed on the query design for the authentication and search pages.

3.1.2 PHP

The first phase of web-page construction was to build the authentication and ordering pages. The layout is based on that of the parent ACLC site. The style section of the home page was copied onto the library pages. Eventually the ACLC site will be converted to use a remote style sheet, but currently the styles are local to each page.

Due to the low traffic on the site, especially at first, testing was a slow process. I had access to computers running Windows 95 and 98, Windows 2000 and Windows XP as well as two running Red Hat Linux 7.3. Browsers available were Internet Explorer 5.5 and 6.0, Netscape Navigator 4.70, 4.79 (two flavors, Win98 and Linux), Mozilla 1.0.1[1], Navigator 6.0 and 7.0, KDE Konqueror, Opera 6.3(again, 2 flavors, Win98 and Linux) and 7.0 for Windows as well as the text-only browser, Lynx. While this was a good range of test browsers, there are many that I was not able to test. Apple has a number of Apple-only browsers such as OmniWeb, as well as versions of Netscape and Opera. I could only hope that I caught the big problems with the PC browsers, and if there were Apple-specific problems, club members with Apple computers would find (and report) them. None were reported.

A problem that appeared after the site was online for at least a month

[1] Mozilla was intended to be an open source "Netscape Navigator 5.0" created by a project started for the most part by Netscape but taking contributions from other companies and independent volunteers, but it became more than just Navigator 5.0 and was released as its own entity. Further proprietary modifications to Mozilla code produced Navigator 6.0 [30] [31]. Navigator 4.79, Mozilla 1.0 and Navigator 6.0 are only slightly related and have quite different characteristics.

was the failure of the order confirmation page. When an order is submitted, the server emails the order to the ACLC Vice President and shows the user a confirmation page to let them know the order was placed. This worked for several weeks, may have sporadically failed on some browsers (or failures may not have been reported) for several more weeks, and finally, around January, failed completely. The order was sent but the confirmation page would not be shown. I found and repaired a minor bug in it about every other week, but due to the low traffic it took most of the Spring semester to track down the actual cause to a problem with the session control. Apparently the database search was interfering with the session, causing it to end before the confirmation page could be sent. Since the semester was about over, and the website will be reimplemented for next year, I "cheated" and rewrote the confirmation page as straight HTML, leaving the PHP sessions "broken" but showing the confirmation to the users.

The amount of display debugging that was necessary for the first release surprised me. As a 'newbie' to web design I understood that browsers were somewhat different but thought that browsers of a comparable level, IE4 and Netscape 4 for instance, understood approximately the same level of HTML, Netscape being more unforgiving about unclosed tags and such. However, even the same version browser on different versions of Windows can respond differently. A great deal of time can be spent fixing the result of previous fixes which now caused other problems in a different browser.

When different operating systems are added to the mix the results are even more unpredictable. For instance, the style sheet sets the color of the text. This worked fine in Internet Explorer and (mostly) correctly in Netscape Navigator 4.79 in Linux. However Netscape Navigator 4.79 in Windows 98 had problems with the style sheet, and text color had to be set with `<font>` tags. Luckily the tags did not cause retroactive problems in the browsers that handled style sheets properly.

After the basic log on and order pages were functioning (more or less) reliably, I added the database search functions. Then I expanded the database search functions so users could search for tapes by language as well as by title. Next, I added more user-selectable search parameters, to allow search by language, medium, and/or release type, and to order the result by ID number or alphabetically by title. This is the final implemented version.

The next phase calls for more tables, such as DIRECTOR and STUDIO, and consequently more complex queries. Also, Web-based database modification so that members can sign themselves up, and officers can modify the library database over the internet is needed. This will require more advanced authentication and security, since real data will be stored and modified. Since member's personal information and club records would be accessible on the internet, some reliable method would have to be designed to make it only available to the "right" people. phpBB is a "high powered, fully scalable, and highly customizable open-source bulletin board package" based on PHP, that has these features built in. It is intended for hosting Web Forums and has authentication features appropriate to that. It could be modified to provide an ordering system in addition to the self-sign-up and variable posting privilege features it already has [32].

3.2 SPG

The original plan for this website was abandoned after the first draft was produced and implementation proceeded in a thoroughly disordered fashion after that. I tried to keep the look of the pages coordinated with a template, but some pages required that design to be modified slightly.

3.2.1 Database

The State Planning Grant has the financial resources to purchase proprietary "closed source" software. They are also able to share resources, such as computer and network hardware, Microsoft licenses, and other software with the Center for Rural Health Research and Education (CRHRE). In the case of the public Web site this includes the *health.uwyo.edu* server, it's Windows 2000, and later Windows XP, operating system with ASP and IIS, and Microsoft Access database. Eventually, the database will be upgraded to SQL Server, for which CRHRE also owns licenses.

3.2.2 ASP

Active Server Pages is the Microsoft script processor. It handles VBScript and JScript. A version for use on Unix/Linux servers is available, but all scripts for it must be written in Perl [24]. ASP is a powerful server-side processor with easy database connectivity through ODBC. The same ODBC commands are used to connect to either Access or SQL Server, so the database can be upgraded without rewriting the Web scripts [21].

3.2.3 Microsoft

SPG uses Microsoft and other proprietary software both because they have the finances available to do so and for the support and consistency. Concurrent versions of MS software tend to uniformity of interface design, which offers a more consistent environment from product to product than is generally available in free software. I have found there is usually more than enough time spent trying to remember "where was that button again?" in a consistent GUI design without adding "*is* there a button, or do I type a command?" to the mix.

3.3 Scalability

PHP/MySQL is ready for mid-size corporate web traffic. MySQL accepts 101 simultaneous connections. Currently, only Microsoft SQL Server offers more simultaneous connections than MySQL [33]. POSTgreSQL would not add any speed, as MySQL is faster on the same hardware. POSTgreSQL does have features that MySQL does not, such as full support of transactions, triggers, full joins, etc. See Table 3.1 for details.

ASP will handle that much traffic, Access will not; Access is a personal database program, not a database server [21]. Even though Access is more than capable of handling the ACLC and SPG search traffic, about 20 to 30 searches per week. Access could be replaced with SQL Server which should function with the search pages without requiring any changes to the code [21]. Thus the software behind both the ACLC and SPG sites is capable of handling more traffic than either (or both combined) will probably ever face.

The ACLC library "server" is pretty much obsolete. As a dual processor machine it has more power than a single Pentium II could provide, but it is still a Pentium II machine. It could also benefit greatly from more memory; 256MB is not enough to run today's graphically intensive programs, free *or*

	POSTgreSQL	MySQL
ANSI SQL compliance	Closer to ANSI SQL standard	Follows some of the ANSI SQL standards
Performance	Slower	Faster
Sub-selects	Yes	No
Transactions	Yes	Yes, however InnoDB table type must be used
Database replication	Yes	Yes
Foreign key support	Yes	No
Views	Yes	No
Stored procedures	Yes	No
Triggers	Yes	No
Unions	Yes	No
Full joins	Yes	No
Constraints	Yes	No
Windows support	Yes	Yes
Vacuum (cleanup)	Yes	No
ODBC	Yes	Yes
JDBC	Yes	Yes
Different table types	No	Yes

Table 3.1: MySQL and POSTgreSQL comparison, from Hunter [2]

proprietary, at a comfortable rate. I also used it as my office workstation and while I never noticed a lag in Web-based performance[2], I did notice considerable lag in desktop performance at times. If the ACLC were to expand the library functionality significantly, or host the entire site with its considerably higher traffic on this machine, *wks16.cs.uwyo.edu* would not be up to the task.

SPG shares space on the *health.uwyo.edu* server with several other CRHRE-related websites. The online speed is again acceptable, while the desktop speed sometimes lags substantially.

[2] In raw bits-per-second throughput, even a single original Pentium I processor can send and receive data faster than the data can travel the network: 75MHz processor speed versus 1.5Mbps(T1), 8Mbps(DSL), 10Mbps(LAN). The big bottle-neck is read/write operations at the hard-drive, and running the program that was just downloaded [34]

3.4 Security

The ACLC site uses an input form, hiding the user's passwords with the HTML password tag. This information is passed to a PHP script which checks the username and password against the member database to screen orders. No online database modifications are possible at this time; the only reason for any security is to prevent (hypothetical) orders from non-members, or from people who think that ACLC sells videos. Access to the database itself is controlled by authentication on the server and a MySQL login. Access to the video library is controlled by the Vice President. Videos are generally handed out in person at the meetings, so membership status can be determined at that time. Non-members who somehow logged on to the order-form and ordered a video could pay dues "on the spot" and receive the video. Most members will not be ambitious enough to crack this and will just send an email requesting a tape. They will have to be verified personally by the Vice President either way.

The SPG site does have online modification of the resources database. The modification page is not linked from any publicly available page and has a JavaScript password checker to redirect unauthorized users. To make changes through the modification page, a user would have to know that the page exists and what the address is. The name of the page was chosen to be as hard to guess as possible: it is not "changeDatabaseHere.asp." If someone can find the page, they would only have to view the source code to read the password, as it is not really hidden. A user who is already logged on to the server could easily find the password, but if they are on the server, they already have enough access to make changes without the password. Also, the database contains nothing more critical than a list of publications and websites for public information.

Chapter 4

Evaluation of Project

4.1 ACLC Project

4.1.1 Software Engineering

I used a spiral model to plan the ACLC project. Each phase was completed before the next was begun. Many of the tables had been designed as a semester project in Database Systems class. They were originally implemented in Oracle. The basic design did not need to be changed, though some of the original tables were not used, and new ones, built on the same pattern, were added. Only minor modifications of the SQL command scripts had to be made to work with MySQL's syntax.

I used the style sheet and layout of the pre-existing ACLC website built by Darren Bills to make the login and search pages consistent with the look of the rest of the site.

The features of the new pages were designed by Darren Bills, ACLC President, and me, ACLC Vice President. Later in the process, Chris Lange, the incoming President made some suggestions for modifications.

4.1.2 Miscellaneous

I had near-total control of my part of the ACLC page. Almost any suggestion I made regarding the database was accepted: as long as I implemented it, no one was going to talk me out of doing free work for the club. As Vice President I had the authority in the club to make changes affecting the library, and no one else was interested in working on the project.

4.2 SPG Project

4.2.1 Software Engineering

We did some planning for this site ahead of time, but not much. The original plan was to base the site on the version that was already posted, first unifying the design, then incrementally improving it and adding features. The SPG staff felt that it was more important to have something up immediately than to wait until it was perfect.

Site layout was discussed at the start of the project, but the original look turned out to be not what they really wanted. After that changes were apparently made on the basis of "I'll know it when I see it." The production schedule changed as priorities were rearranged. Many incompatible design concepts were required. For instance, headings in text content "had" to be on a single line, with no wrapping, but the entire page had to print in portrait orientation without cutting off the right side of the page, or forcing the user to make any settings changes on their computer. Since the W3C accessibility standards require allowing the user's browser or other reader to choose font and width settings [35], using tables to force the page to a printable width was deprecated but necessary.

4.2.2 Microsoft-based

Because the SPG already has Microsoft programs, and to make maintenance less intimidating for "non-geeks," I used them for the site. Scripting for the contact form and for database searches was done in ASP, using VBScript. I used JavaScript for all scripting not involving database access. All editing was done with "non-encoding" editors such as Notepad, 1stPage 2000, and occasionally Microsoft Visual InterDev, which has error messages and other helpful features but allows text editing without "encrusting" the text with unnecessary code.

According to the books, the Access database may be moved to Microsoft SQL Server later and the ODBC code can remain unchanged [21]. However, I have found in the course of this project that "standard" code often isn't, and small tweaks will probably have to be made.

4.3 Questionnaire

4.3.1 Discussion of Issues

To get more measurable feedback on the success or failure of the ACLC database, I wrote a short questionnaire. Since I am not a statistician, and the numbers involved are so small, this survey should be treated more as a focus group and not as a real survey. I presented the questionnaire at one Friday night meeting in late April. Of the 20 paid ACLC members, 10 were present. However, I excluded myself as the author of the system under study.

One non-member was also present. As 9 out of 20 eligible members were present, numbers are possibly significant. Of the 4 members who regularly checked out tapes, 3 participated in the survey, which may skew the results statistically, but give a good impression of the general reception to the design. The summary of the results is given in Table 4.1.

	Yes	No		
Member?	9	1		
Use Site?	9	1		
Use Database	7	3		
Check Out Tapes?	5	5		
How Often?	Weekly	Monthly	Once or Twice	I can check out tapes?
	3	0	2	2
How do you request tapes?	In Person	Email	Website	
	1	1	4	
What do you like about the site?	Design	Convenience	Database Search	Other
	2	5	3	0
What do you dislike about the site?	Design	Inconvenience	Database Search	Other
	2	0	1	3

Table 4.1: ACLC Questionnaire Results

The database/ordering system itself received few negative comments. For unspecified reasons, 2 respondents disliked the design, one disliked the database search. Of the 3 written 'Dislike-Other' comments none regarded the database/ordering system itself. One disliked 'nothing', one disliked 'frames' (which are a common cause for complaint about the general site design as they make it appear too cluttered and 'busy'). One respondent remarked that the library contained too few videos, DVDs in particular.

4.3.2 Opinion

All of the software, free or proprietary (even Access), can handle much more traffic than the sites are currently receiving. Some, like MySQL, are designed for vastly greater use, bigger, more complicated tables and more complex queries. Papers on load testing of websites [36] [37] give figures at least 2000 times higher; 3 hits per second rather than 20 - 25 hits per week. So I can not claim to show any great findings about relative speed of one program or scripting language over another. What I can say is that in my opinion, software engineering principals have value in even the smallest projects.

It would have been easy enough to "whip something up" for the Anime Club, but since the planning had been done ahead of time, I could get the important functions implemented and then concentrate on fixing small but hard to find problems like the malfunctioning order confirmation page, and other small display fixes. The SPG project, on the other hand,was characterized by constantly shifting priorities. Several times, the functionality of a page would become secondary to a cosmetic change.

The concept of requirements triage [38] is useful. Once the requirements are determined, they may be prioritized. Some must be done immediately, some can wait till later. Still others may be totally optional, nice to have if there is time to implement them, but not necessary to the functionality or utility of the project. Having a schedule, even if I did not stick to the original time table because of the slowness of testing, let me approach the ACLC project in a step-wise manner: "finish this part, then move on to the next."

Chapter 5

Summary and Future Work

5.1 Summary

The ACLC project was, in many ways, an ideal project, or a laboratory exercise. I was the designer, programmer, and part of the client organization. Once the original design work was done, no one in the club wanted to be involved in the programming, so I was left alone to build it as I saw fit, with whatever programs or languages I chose. It was a small project, so one person could easily do the work: no teamwork with its interpersonal difficulties was required.

The SPG project was very much in the "Real World." I had clients to satisfy, time tables (fairly liberal, but not infinitely elastic) to deal with, a predetermined suite of software to use, and other people to work with. Because of the shifting priorities, almost everything had to be done as fast as possible, and could be abandoned at any time (and required again, later).

5.2 Future Work

5.2.1 ACLC

Several tables and the associated queries remain in the design document that have not yet been implemented. The entire ACLC site needs to be converted to a new, cleaner, less cluttered look, using cascading style sheets rather than frames and tables for formatting. The current manual entry of member names and passwords each semester needs to be replaced with a self-sign-up Web page where members may enter their own information. A Web-based modification interface to allow changes to the library database to be made without logging on to the server and writing SQL should be added.

The program 'mysqldump' extracts all tables and data so the database can be moved as a plain text file and recreated on a new server. Since the generated file contains SQL commands, it can be used with most if not all database programs, including Access, if that is the only thing available [16].

If no server is available for the PHP scripts the web pages will have to be rewritten for client-side operation so they may be hosted on IT's computers. One way to do this is to build the database in JavaScript, another would be to convert it to XML. Either would require more work and skill than most ACLC members are able to provide. However, the new President of ACLC, Chris Lange, has server space as well as considerable MySQL/PHP skills. He has expressed interest in converting the database/ordering system to phpBB [32], a new software package that will allow easy implementation of the advanced features of the design.

I have avoided XML for this project in the interest of simplicity. To make this project as non-threatening as possible for the next ACLC webmaster/librarian, who may not even be computer literate, much less a "geek," I have avoided all advanced functionality. The code is as straight-forward as I was able to make it, and the display is thus rather crude. I imported the inline style sheets Darren Bills used for the other pages in the site and avoided anything beyond very basic HTML/PHP code. Part of Software Engineering is planning for maintenance, and the next maintainer of this code may need to work on these pages with one hand on the keyboard and the other hand holding the PHP/MySQL manual. While there is a high correlation[1] between anime fandom and computer savvy, they are not guaranteed to go together. Also, the person willing to take over the library may not have any web-coding ability. XML might simplify implementation of some planned features, but I felt that it was more important to keep it as basic and simple as possible. The new (2003 - 2004) President of the ACLC, Chris Lange, has a Web-hosting business and extensive Web-programming experience. He will, if he is interested and has the time, be able to finish implementing my plan, or design and build his own. I know that he intends to keep the site in PHP/MySQL, and he may add XML. The entire ACLC site, not just the database portion, will receive a complete cosmetic overhaul. Chris also intends to add the graphical front-ends that are needed to make the MySQL administration more accessible to the average non-computer scientist who is likely to get the Vice President position in the future.

[1] "Geek(modern definition) A person that is into technology, especially computing and new media." [39]

"There are many different types of people who watch anime, but the most common anime viewers come from one of 3 core groups. These groups are college students, teenagers and self-titled "techno geeks." [40]"

5.2.2 SPG

The SPG site will not change much. It is for the most part done, and only a few periodic changes in content (not form) will be made. My part in it has ended, now that the main design and coding phase is over. I tried to make the design as modular as possible as an aid to maintainability. The Web pages are broken into sections, with comments that I hope are self-explanatory, and all the scripts are fairly small and simple. The only major change yet to be made is to move the database from Access to SQL Server.

5.2.3 Further Comparison

This project studied a free software-based, engineered website and compared it to a proprietary Microsoft software-based, largely unplanned site of roughly the same size and nature. A more thorough comparison of the software could be achieved by using free software on an "organically grown" site, and Microsoft products and software engineering methods on another, equivalent, site. To have even better control of variables, four similar or identical websites should be created: 1) free software, engineered, 2) free software, organic, 3) Microsoft, engineered, and 4) Microsoft, organic. The team composition should be matched as closely as possible.

Chapter 6

Appendices

6.1 Design document

6.1.1 Anime Database Project, Requirements Document

```
Things the ACLC Database must do:

Track tapes - CheckedOut, History

Track anime - Actor, Character, Studio, Director

Track members - Officer?, Has Tape?
```

Online ordering - members can search library, request one tape.

May request second/third choices

If a member has a tape checked out they can still request a new one but they must return the old one, can swap at the meeting

Security - officers (Pres, VP) can add delete tapes/anime

Can mark tapes as "Pulled"

(Pres, Secretary) can add delete members

Ordinary members can request tapes but cannot change the database

General Requirements

The database will be constructed using MySQL on a Linux computer.

Web access will mainly be handled using PHP and a combination of HTML, Java, Perl, C/C++. Whatever works and is free will be considered.

The DB must be simple to use, as members will be interested in searching the ACLC's Anime Library, not in learning an interface. The DB must be simple to maintain, as future librarians cannot be assumed to be computer literate, much less able to write

SQL code. If the code can be easily ported from Linux to Windows, the ACLC would have a better chance of finding a server to run it on, if the COSC Department will not allow it to reside permanently on one of their servers. A ready to run binary would be nice. Either server-side processes must be eliminated (hard to do since PHP is a server-side application) or the IT Department must be convinced to allow it to run on UW computers (this is not an issue as long as it runs on COSC machines. [I think.]).

Specific Requirements

If names can be entered in one box (John Smith), then searched as separate fields ([John], [Smith]), we can avoid the confusion of which is "first name" and which is "last name" since Western and Japanese order are both used (rather indiscriminately) on anime credits. This will probably require searching for each name in both first and last name tables [Name1 = "John" or Name1 = "Smith" and Name2 = "John" or Name2 = "Smith"] (May replace 'and' with 'or'. This should also make it possible to search in cases where only one name is known.).

Anime Club Laramie Chapter

Library Database

Description of Database

The Anime Club Laramie Chapter maintains a library of anime videos that members can check out. This is a database for tracking and maintaining that library.

Each tape is given a unique tapeID number. When a tape is checked out, the tape table is updated to show that the tape is checked out. When it is returned, the table is updated to show the tape is available. Various characteristics of the tapes and the animes on them are maintained, so that users can find animes that are suitable to their tastes. Viewers can find answers to the following sorts of questions.

Possible Queries

1. Which animes featuring the voice of Megumi Hayashibara are not checked out?

2. How many animes were made by the Gainax studio?

3. Which animes made by the Gainax studio were directed by Hayao Miazaki?

4. How many copies of Castle of Cagliostro are not checked out?

5. Which animes feature the character of Lupin III?

6. Which characters are played by more than one actor?

7. How many times has Hyperpolice been checked out?

8. How many dubbed tapes from studio Ghibli are in our library?

9. Which actors play more than one character in the same anime?

10. Which animes are directed by Mamoru Oshii and feature the voice of Mimi Woods?

11. What date was an anime released?

12. Is <Fansubbed anime> still available, or has it been pulled from the library?

13. What is the most popular tape?

Assumptions

* All tapes are in color.

* "Anime" = movie or series episode(s).

* If anime.ReleaseType = "Fansub" and tape.Rights = Bought"

Then tape.status is set to "Pulled" and the tape can not be checked out. tape.Rights = "Bought" for all movie.ReleaseType = "Commercial".

* "member" and "member-tape" tables allow tracking of which members have checked out which tapes, and who is allowed to edit the database.

* Members may or may not have ID numbers.

* Members may only check out one tape at a time.

* Members will be able to request tapes from the Web page. Requests posted before 4:00 pm on club meeting days will be filled at the meetings. If members want tapes at other times they will need to make arrangements with the Vice President.

* All tapes in the library are in VHS format, and all tapes contain anime.

* Some tapes might contain more than one anime (multiple episodes). The same anime may appear on more than one tape (subbed and dubbed copies).

* It is possible for the same anime to have more than one title (FuriKuri = FLCL = Fooley Cooley). It is possible for the same title to apply to more than one anime (multiple episodes).

* The same voice actor may play more than one character, even in the same anime. The same character may be portrayed by more than one actor, sometimes even in the same anime (Saotome Ranma).

* More than one studio may participate in making the same anime. One studio may make several animes.

* It is possible that two directors could colaborate in making the same anime. The same director can make more than one anime.

* No two directors will have the same name. No two studios will have the same name.

Notes

* "Fan Service" is a term used to describe the sex and violence content of a anime. In this DB, it takes the form of the fsclass table, wherein several categories, including sex, violence, and others, are each rated on a scale from 1 to 5.

Relational DB Schema

Entities:

tape(TapeID,Status,Rights,History)

Status domain: Checked Out, Available

Rights domain: Bought, Not Bought

History: How many times has this tape been checked out?

anime(AnimeID, TitleID, Language, ReleaseType, FSClassID, ReleaseDate)

Language domain: Subbed, Dubbed

ReleaseType domain: Fan Sub, Commercial

title(TitleID,AnimeTitle)

actor(ActorID, ActorName1, ActorName2)

character(CharID, CharName1, CharName2)

studio(StudioName, Location)

director(DirectorID, DirectorName1, DirectorName2)

fsclass(FSClassID, Sex, Violence, Profanity, Nudity, Gore)

Domains: All categories rated on scale from 1 to 5

member(MemberID, MemberName, Officer?)[Name1, Name2?]

Not all members have a number [may have to require numbers]

Officer? == Boolean. Only officers are allowed to edit the database

Relations:

anime-tape(AnimeID,TapeID)

anime-title(AnimeID,AnimeTitle)

anime-dir(AnimeID,DirectorName)

anime-stud(AnimeID,StudioName)

actor-char(ActorID,CharID,AnimeID)

member-tape(MemberID, TapeID)

Functional Dependencies

tape F = {TapeID -> {Status,Rights,History} } BCNF

anime F = {AnimeID -> {TitleID, Language, ReleaseType, FSClass, ReleaseDate} }

title F = {TitleID -> AnimeTitle} BCNF

actor F = {ActorID -> ActorName1, ActorName2 } BCNF

character F = {CharID -> CharName1, CharName2} BCNF

studio F = {StudioName -> Location} BCNF

director F = {DirectorID -> DirectorName1, DirectorName2 } BCNF

fsclass F = {FSClassID -> {Sex,Violence,Profanity,Nudity,Gore} } BCNF member F = {MemberName -> MemberID, Officer?}

Decomposition

No further decomposition is necessary. All relations are in BCNF.

6.2 Statistics

6.2.1 Example Statistics

ACLC Sitemeter 6/23/2003

VISITS

Total 64 987

Average Per Day 71

Average Visit Length 0:23

Last Hour 4

Today 48

This Week 496

PAGE VIEWS

Total 113,017

Average Per Day 97

Average Per Visit 1.4

Last Hour 7

Today 88

This Week 677

Plus 4,572 visitors before joining Site Meter on June 1, 2001

SPG Sitemeter 7/03/2003

VISITS

Total 949

Average Per Day 6

Average Visit Length 5:14

Last Hour 0

Today 2

This Week 44

PAGE VIEWS

Total 2,686

Average Per Day 11

Average Per Visit 1.7

Last Hour 0

Today 5

This Week 74

since Nov 5 2002

6.3 Questionnaire

6.3.1 ACLC Website Survey

1.5

Are you a member?

Yes No

Have you used the ACLC website?

Yes No

Have you used the ACLC library database?

Yes No

Have you checked videos out of the library?

Yes No

How often do you check out videos?

Weekly Monthly

Once or twice I can check out videos?

How do you usually request videos?

In person Email Website

If you use the website to order videos, what do you like best about it?

Design Convenience Database search

Other ____

If you use the website to order videos, what do you like least about it?

Design Inconvenience Database search

Other ____

Bibliography

[1] Roger S. Pressman. *Software Engineering: a Practitioner's Approach.* McGraw-Hill, 5th edition, 2001.

[2] Shylon Ray Hunter. MySQL vs. PostgreSQL. `http://builder.com.com/article.jhtml?id=u00320020624gcn01.htm`.

[3] Darren Bills. ACLC Website. `http://www.uwyo.edu/anim`.

[4] Eric Raymond. The New Jargon File Website. `http://www.catb.org/~esr/jargon/html/C/copious-free-time.htm`.

[5] SPG Public Information Website. `http://health.uwyo.edu/spg/`.

[6] Jim Osborn. UW Information Technology Department, Server-side Policy. Personal Communication, 2002.

[7] Software in the Public Interest Website. http://www.spi-inc.org/projects/opensource.

[8] Opensource.org Website. http://www.opensource.org.

[9] Laura A. Majerus. The Open Source Software Movement and Intellectual Property. http://www.fenwick.com/pub/ip_pubs/IP_Bulletin_Summer_99/ip_bulletin_summer_99.htm, Summer 1999.

[10] Richard Stallman. The GNU Project Website. http://www.gnu.org.

[11] Various. Various advertisements. *Network World*, July 14 2003.

[12] Dell Computer Corporation Website. http://www.dell.com.

[13] Peter Galli. Sun's Linux plan extends support. *eWeek*, April 7 2003.

[14] Linux.org Website. http://www.linux.org.

[15] Corel Website. http://www.corel.com.

[16] Luke Welling and Laura Thomson. *PHP and MySQL Web Development*. Sams, 2001.

[17] Java Sucks Website. http://www.internalmemos.com/memos/memodetails.php.

[18] Microsoft Frontpage Website. http://www.microsoft.com/frontpage/.

[19] 1stPage 2000 Download Website. http://www.freewebmasterhelp.com/reviews/software/firstpage.htm.

[20] Evrsoft Software Website. http://www.evrsoft.com/download/.

[21] John Kauffman, Kevin Spencer, , and Thearon Willis. *Beginning ASP Databases.* Wrox Press, 1999.

[22] Mitch Tulloch and Patrick Santry. *Administering IIS 5.0.* McGraw-Hill, 2000.

[23] Windows Server 2003 Website. http://www.microsoft.com/windowsserver2003/techinfo/overview/iis.mspx.

[24] Apache-ASP Website. http://www.apache-asp.org.

[25] OReilly Publishing Website. http://www.oreilly.com/catalog/perldbi.

[26] Openlink Software Website. http://www.openlinksw.com.

[27] Anime Turnpike Website. http://www.anipike.com.

[28] Nicholas Duarte. Anime on Campus. *NewTypeUSA*, 2(6):143, June 2003.

[29] ACLC Sitemeter Website Statistics. http://sm7.sitemeter.com/stats.asp?site=sm7crant.

[30] Mozilla Project Website. http://www.mozilla.org/docs/mozilla-faq.html.

[31] Mozilla/Netscape information Website. http://www.gerbilbox.org/newzilla/general/general02.php.

[32] phpBB Website. http://www.phpbb.com.

[33] Mark Maslakowski. *Teach Yourself MySQL in 21 days.* Sams, a division of Macmillan USA, 2000.

[34] Jr. Frank J. Derfler. *Practical Networking.* Que, 2000.

[35] Web Content Accessibility Guidelines 1.0, W3C Recommendation 5-May-1999. http://www.w3.org/TR/1999/WAI-WEBCONTENT-19990505/.

[36] Ian Gorton, Anna Liu, and Paul Brebne. Rigorous Evaluation of COTS Middleware Technology. *IEEE Computer*, 2003.

[37] Martin Arlitt, Diwakar Krishnamurthy, and Jerry Rolia. Characterizing the Scalability of a Large Web-Based Shopping System. *ACM Transactions on Internet Technology*, 1(1):44–69, August 2001.

[38] Alan M. Davis. The art of Requirements Triage. *IEEE Computer*, 2003.

[39] Wikipedia Website. `http://www.wikipedia.org/wiki/Geek`.

[40] Phrozen-Neon Website. `http://www.phrozen-neon.com/apoll`.