IPRO-357 Summer 2003
Strategic Management System Software
System Redesign Team's Midterm Progress Report

By
Casey Ligas
Go Nakagawa
HuanKiat Tang

June 26, 2003

Abstract

This midterm report contents the details of what System Redesign team has accomplished on the software, what we plan to improve for the software. This midterm report further elaborates the assumptions that we had made in the project plan, and the answers to the questions brought up in the project plan. Some technical terms (e.g. SQL, PHP) are added with explanations to allow the readers to understand the elaborations.

Table of Content

Introduction

Balanced Scorecard

"As companies around the world transform themselves for competition that is based on information, their ability to exploit intangible assets has become far more decisive than their ability to invest in and manage physical assets. Several years ago, in recognition of this change, we introduced a concept we called the balanced scorecard. The balanced scorecard supplemented traditional financial measures with criteria that measured performance from three additional perspectives – those of customers, internal business processes, and learning and growth. It therefore enabled companies to track financial results while simultaneously monitoring progress in building the capabilities and acquiring the intangible assets they would need for future growth."
(From *Using the Balanced Scorecard as a Strategic Management System*, by Robert S. Kaplan and David P. Norton, Harvard Business Review January-February 1996)

Strategic Management System Software

"The Strategic Management System Software Version 1.0(SMS) was developed as a technology commercialization system used to create situation analysis, business strategies, and operational feasibility analysis. The SMS software helps build tools and techniques to manage across six different perspectives: vision, mission, financial, internal business process, customers, and learning & growth."
(From *IPRO-327 Project Plan Summer 2003*)

The SMS software is a web-based system that uses the strategy map model developed by Dr Kalidonis that integrated with Balanced Scorecard. Currently a newer version 2.0 is in development.

Web Trend

As the technology advanced in last decade, the understanding of the Information Technology (IT) has been changed. Internet was one of the major advancement that changed the concept of retrieving information. Now almost any information is available on the internet. Also the number of people using internet is increasing constantly while internet service provider is also competing against each others to provide more comfortable and sufficient services. We decided to develop our software, SMS 1.0 and 2.0, on the web based system so that the customers have access to the software anytime.

Module Overview

- Analyze and critique the current design architecture
- Identify and present a series of alternative design architectures
- Make a series of recommendations to improve future versions of the SMS software and build a new version of the SMS system.

Objectives/Milestone and Expected Results

Vision/Mission

Our goals are quite simple. We aim to develop a redesigned user-friendly and more optimized system in which the server can execute the ASP code efficiently so that the output of the code will be sent to the client computers faster and much effectively. Through system evaluations, redesign and code optimization, the Strategic Management System Software Version 2.0 will rise above its predecessor in graphical interface and functionality to bring solutions to small and medium family businesses. The solutions includes: situation analysis, business strategies, and operational feasibility analysis.

Elaboration of Specific Questions

Which platform would SMS software perform better? UNIX or Windows based?

Let us look at the comparison table below to support our decision that Windows based servers are more effective to improve the current version of SMS.

Table 1

	UNIX Based	Windows Based
Pros	1. IIT servers are mostly UNIX based 2. Plug-in is available for ASP 3. More Stable	1. Widely used 2. SMS is Windows based 3. The team can focus more on improving the system
Cons	1. Time Consuming 2. Database Compatibility	1. Cost us money 2. Less Stable

UNIX Based Pros:

1) Since the servers at IIT are UNIX based, we would not have to pay for hosting at the present time and we would have the same functionality as a pay service.

2) Currently the ASP code is not supported by the UNIX based server. However, *Chilisoft, the software company, has developed a plug-in to emulate the ASP platform to execute the ASP code (www.chilisoft.com 06/24/03).

*Chilisoft
"…a cross-platform implementation of Active Server Pages (ASP) that allows Java and ASP developers to work collectively on developing portable ASP applications for a variety of platforms…

…Sun Chilisoft provides the bridge for connecting the Java platform to ASP code for cross-platform deployment of ASP applications…"

(Source: http://www.sun.com/smi/Press/sunflash/2002-03/sunflash.20020326.2.html)

3) It is a well known fact that the UNIX based server is more stable than the Windows based server. Most corporation use UNIX servers because of their stability. Here are some of the reasons:

Table 2

UNIX	Windows
UNIX system's uptime can be measured in years. Reports of uptimes reaching 3 years are not uncommon in the Linux community.	The infamous "*Blue Screen of Death" can happen anytime, daily, weekly, or monthly, depends on how it's used, and sometimes it's unpredictable.
Most large corporations are UNIX-oriented, they normally go with Web server software like Apache or Netscape-Enterprise.	Microsoft's IIS Web server software does not even amount to one-quarter of all Internet-connected Web servers.
The famous free web-based email server "Hotmail" owned by Microsoft is running under UNIX.	Windows based servers are not capable of handling the amount of requests from the increasing number of Hotmail's users, and that can only be done under UNIX based servers.

*Blue Screen of Death
Blue Screen of Death is a situation in which the normal desktop window system disappears completely and is replaced by a full screen of hexadecimal numbers and/or error messages on a blue background. The *only* method of recovery in this situation is powering the machine off and rebooting.
(sources: http://www.linuxfocus.org/English/May1998/article41.html and http://geodsoft.com/opinion/server_comp /summary.htm)

UNIX Based Cons:

1) We have found the alternative coding (*PHP) that is platform independent (You can host the software regardless of what operating system the server is depending on; UNIX, Windows, Mac OS, and etc.). From our point of view, it is required to change the entire ASP code into PHP code to ensure the software platform independence. The transition from ASP to PHP will be very time consuming because we will have to rewrite the SMS software in PHP code and we are not yet familiar with the scripting language itself.

2) Currently, SMS version 1.0 uses Microsoft Access database. Unfortunately, PHP does not support Access or *SQL Server (**PHP for further details) at the present time.

*PHP
"PHP (recursive acronym for "PHP: Hypertext Preprocessor") is a widely-used Open Source general-purpose scripting language that is especially suited for Web development and can be embedded into HTML…

…PHP can be used on all major operating systems, including Linux, many Unix variants (including HP-UX, Solaris and OpenBSD), Microsoft Windows, Mac OS X, RISC OS, and probably others. PHP has also support for most of the web servers today. This includes Apache, Microsoft Internet Information Server, Personal Web Server, Netscape and iPlanet servers, Oreilly Website Pro server, Caudium, Xitami, OmniHTTPd, and many others…

…So with PHP, you have the freedom of choosing an operating system and a web server…"

**PHP

Database supported by PHP are: Adabas D, dBase, Empress, FilePro (read-only), Hyperwave, IBM DB2, Informix, Ingress, InterBase, FrontBase, mSQL, Direct MS-SQL, MySQL, ODBC, Oracle (OCI7 and OCI8), Oyrimos, PostgreSQL, Solid, Sybase, Velocis, UNIX dbm

(Source: http://us4.php.net/manual/en/)

*SQL
"SQL stands for Structured Query Language. SQL is used to communicate with a database… it is the standard language for relational database management systems. SQL statements are used to perform tasks such as update data on a database, or retrieve data from a database. Some common relational database management systems that use SQL are: Oracle, Sybase, Microsoft SQL Server, Access, Ingres, etc. Although most database systems use SQL, most of them also have their own additional proprietary extensions that are usually only used on their system…"

(Source: http://sqlcourse.com/intro.html)

Windows Based Pros:

1) From the statistics we found (refers to Assumptions section), Windows is still the widely used operating system for home-end computer users. The existing version of SMS software is hosted on a Windows based server, and the clients will not have compatibility problems to access the software.

2) Spring 2003 team had already developed SMS 1.0 with ASP code which is more reliable on Windows based server because ASP is developed by Microsoft, and is designed to run under Windows platform. In order to integrate well with ASP codes in SMS software, it is better to run the software in Windows based server.

3) By keeping the software on a Windows platform using ASP, we will not be required to rewrite the code to make it compatible with a UNIX server. Thus, we can focus on the system redesign and use most of our time for the implementation of SMS version 2.0.

Windows Based Cons:

1) Hosting the software on a Windows server is more expensive than hosting it on a UNIX server. From Table 1, we know that Windows is less stable than UNIX. It requires more resources to make sure the Windows based server is operating properly.

2) Refers to Table 1.

How well the existing code can be implemented with the latest technology in the future?

The ASP 3.0 platform will be easily updated to an ASP .NET platform (which is the latest technology for ASP and it is becoming more of the standard in today's web-based solutions). By adding comments, future classes of this IPRO will be able to easily upgrade the code to ASP .NET.

How can we modify the existing code of SMS 1.0 to lower the time necessary to execute the code on the server side?

By removing unnecessary code and improving the existing code we can limit the amount of code that needs to be executed on the server side. Thus reducing the execution time and increasing the data transmission rate from the server to the client.

For an example, one command in the code takes 1 ns to process. Here is a function that is written to execute by three command lines. It takes 3 ns to process this function. Now, we refine the function, and it can be executed by two command lines and takes only 2 ns to process. It seems no diffrence between 3 and 2 ns, but when the saved time accummulates, it turns out to affect the performance of the software. When it takes less time for the server to process the data, it loads faster in user's computer.

What functionality or design would the customers be more interested in?

- Simplicity of the software
- Visualization of the strategy map
- More useful help file and definitions
- Overall attractive interface of the software as compared to SMS 1.0
- Message board the customers can post their questions and comments
- Quick response/support from the technical support team

Objectives

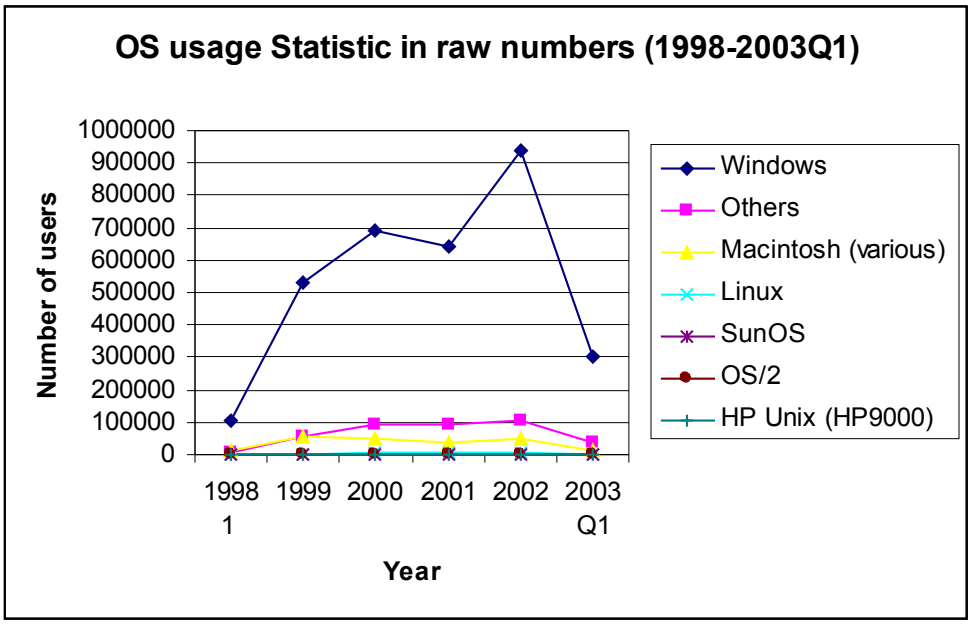
Analyze the current version of system, SMS 1.0, to come up with the ideal version of SMS 2.0 by focusing on the graphical interface, code efficiency, and user-friendly aspects. We also propose the alternative designs of SMS 2.0 to the team so that we can reflect the opinions from them. Discuss with Software Evaluation to get more information and critiques about the other software in the market to improve the current version of software. We will also provide useful recommendations for the next team to accomplish more advanced version of software in the future.

Assumptions

1) Users of the system are using Microsoft Windows as the operating systems for their computers.

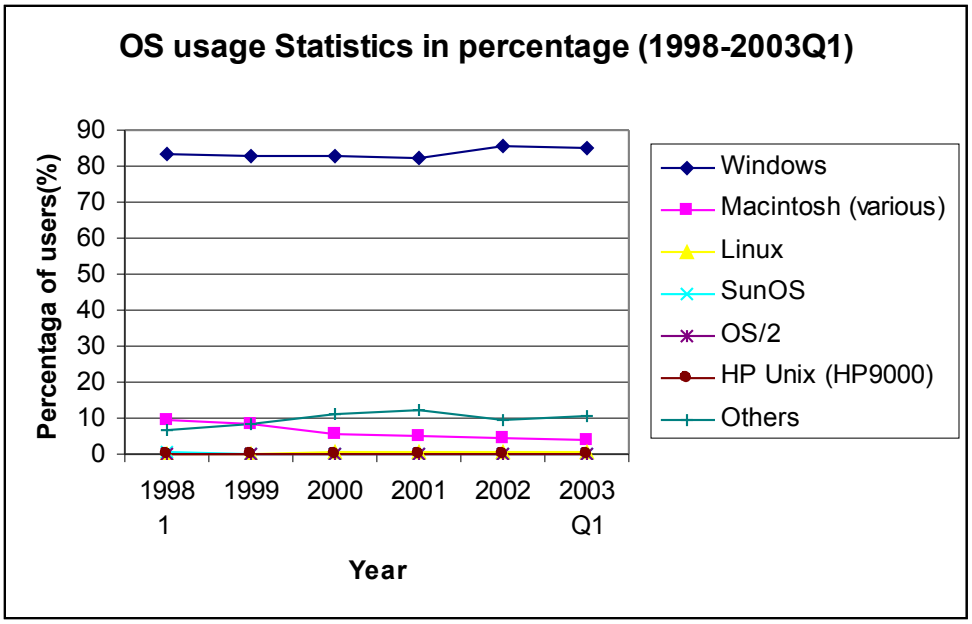
Our assumption that Windows is still the most popular operating system is based on the statistics we found on *http://www.epistemelinks.com/Info/OpSys.aspx*.

Operating System Usage Raw Numbers						
Platform	1998 [1]	1999	2000	2001	2002	2003 Q1
Windows	106561	531965	690872	644270	937270	302684
Macintosh (various)	11678	52528	47787	39296	47603	13774
Linux	314	1501	3531	3132	5148	1554
SunOS	492	1415	1172	857	1003	227
OS/2	88	211	142	63	97	10
HP Unix (HP9000)	71	170	97	36	31	14
Others	8435	53758	90059	94732	102725	36831



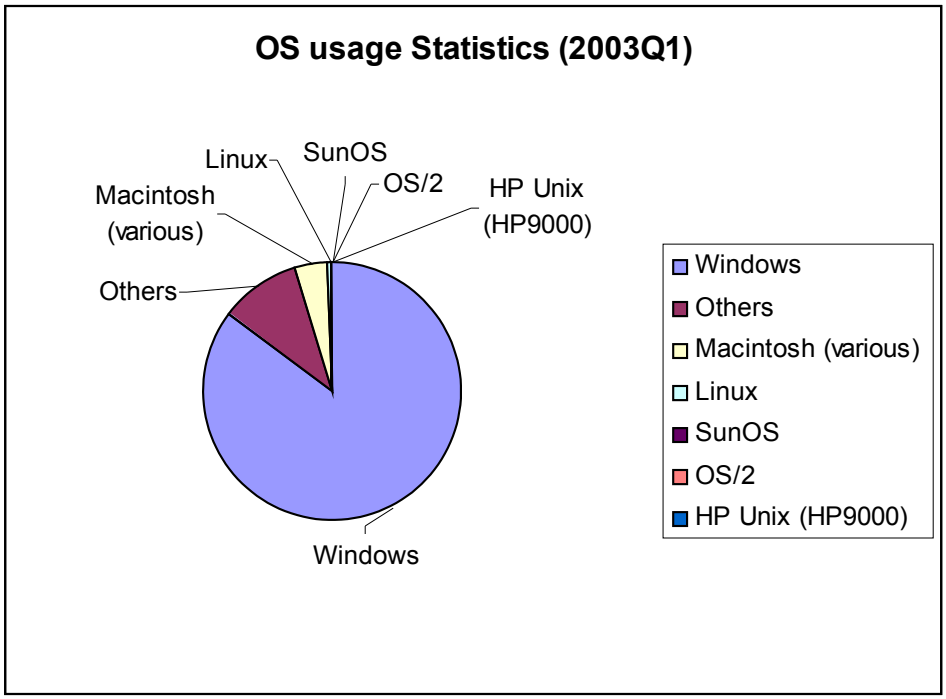
Graph 1: This graph indicates the overall Operating System user sessions or visits for that year (1998 is partial only, and 2003 is Q1 only).

It shows a constant increament of Windows users yearly. But there's a severe drop on year 2003. This confusion is due to the fact that only the first quarter of year 2003 is taken into account. That's why



Graph 2: This graph provides the percentage of the total Operating System user sessions or visits for that year (1998 is partial only, and 2003 is Q1 only).

It shows that Windows is still the majority choice, and it remains almost constantly between 80 to 90 percent until the first quarter of year 2003.



Graph 3: This graph provides the percentage of total Operating System user sessions or visits for the first quarter (Q1) of this year.

**source: http://www.epistemelinks.com/Info/OpSys.aspx*

2) Microsoft Windows users are using either Internet Explorer or Netscape Navigator as their internet browsers.

   These two browsers are the most effective and reliable because they support the latest web technology. From Graph 1, Graph 2, and Graph 3, we know that Windows is the majority choice. Internet Explorer is one of the integrated software that comes along with the installation of Windows. Otherwise, Netscape is the second choice for Windows users.

3) The customers have the connection to the internet with at least a 56K modem.

   SMS is a web-based designed software. We have to make the assumption that the customers that are/will be using our software have an internet connection of at least a 56K modem. Otherwise it is impossible for them to use our software. The major advantage of web-based software is that the customers can access to the software anywhere and at anytime in the world as long as there is an internet connection.

Progress Report

- The latest SMS software is uploaded to the new web server

  The URL of this new site is *http://www23.brinkster.com/ipro357/sms*. This site is independent from the site (*www.stuart.iit.edu/faculty/kalidonis/smsv3.0/default.html*) the other teams are using for the semester (SMS version 1.0). It is for the System Redesign team to try out new functions and new implementations on the software.

  The hosting server for this SMS 2.0 software is free. Some minor changes were made to the unnecessary codes before the files were uploaded to the server.

  The software that the other teams use for the semester is the fully accessible SMS version 1.0 hosted on IIT's server.

- SMS 1.0 evaluation

  Our team has come out with a few alternative design architectures for the software. Improvements and suggestions are needed to decide which architecture will be used for the software.

  We have started to discuss on how to implement the code to make the software compatible with the latest technology.

- Hosting Issue

  We have been researching the prices and the features of where we should host the software. The comparison of prices, features, etc. of the hosting companies should be done by the end of this semester.

Team Overview

Titles

Module	Casey Ligas
Spokesperson	Go Nakagawa
Secretary	HuanKiat Tang

Member Bios

Casey Ligas
I'm a 4th year Computer Science major with interests in networking and programming. I've had experience with this project before (I was in last semesters IPRO and was on the programming team, so I understand the code very well) and I hope to bring that experience to this class. With my background in computer science, which consists of programming, database, and networking, I intend to implement a new and better version of the SMS software, as compared to the previous version.

Go Nakagawa
I'm a forth year Computer Engineer major student. I expect a new experience that would benefit me such as communication skill, leadership role, and teamwork. I currently work for an architecture company, VOA Associates, Inc. in Chicago downtown as a network administrator intern. I have interest in Hardware development as well as network development.

HuanKiat Tang
I am a Computer Engineering Senior. Interested in programming in C/C++, VB/JAVA, and Web Design. This is my second time joining an IPRO project. The first one was about Home-Run measurement system at Comiskey Park. I was on survey team measuring the distances of the ball park.

Activity Schedule and Key Dates

Date	*Activities*	*Output & Deliverables*	*Person in charge*		*Hours of effort*
6/12/03 (6pm-4am)	Upload Software to new Windows based web server	Fully accessible software online	Casey	Uploaded each file to the web server, and made changes to some files so that they are functional on the new server.	10
6/16/03 (7pm-1am)	Analysis of Software	Critiques of the current design architecture (Navigation, interface, code efficiency, user-friendliness)	Casey Go Kiat	Explained the works the last system redesign had done. Gave opinions of what should be improved on the software. Gave ideas of what new functions can be integrated into the software.	6(team work)
6/18/03 (5pm-7pm)	Meet with software evaluation module to discuss the pros and cons of the current SMS software version	Ideas for new design	Casey, Go, Kiat, system evaluation team		2
6/19/03 (6pm-11pm)	Discussion of Midterm presentation	Rough presentation slides of midterm report	Casey Go Kiat	Put opinions and discussions into sentences. Provided thoughts and suggestions to the report Typed and edited the report before it was turned in.	5(team work)
6/23/03	Work on design architecture (new interface, navigation). Work on Midterm report	Rough of midterm report	Casey Go Kiat	Worked on a few new design architectures. Worked on midterm report draft. Worked on midterm report draft.	3 2 2
6/24/03 (7pm-12am)	Work on Midterm Report and midterm presentation slides	Final Midterm report and final midterm presentation slides	Go Kiat	Decided what to be included into the presentation slides. Made slides for presentation, and finalized the midterm report.	5(team work)
6/25/03	Present Midterm Report	Finalized Midterm report and midterm presentation slides	Go Casey Kiat	Looked throught slides for the presentation Refined the content for presentation. Finalized the presentation slides and midterm report.	3 3 4

Date	*Activities*	*Output & Deliverables*	*Person in charge*		*Hours of effort*
6/26/03 (7pm-12am)	Turn in finalized midterm report	Finalized Midterm report	Casey Go Kiat	Proved read midterm report Checked what could be added to the report. Typed and edited the report	10(team work)
6/30/03 (7pm-12am)	Start to implement alternative design architecture that was created during the analysis of the software	SMS 2.0 alpha	Casey Go Kiat	Lead the team in implementation. Studied the codes. Studied the codes.	5(team work)
7/02/03 (6pm-9pm)	Working on alternative designs to the interface of the software	SMS 2.0 alpha interface designs	Casey Go Kiat	Explained how the software was implemented. Came out with several design architectures. Studied how the software functions. Studied how the software functions.	5 3 3
7/07/03 (3pm-5pm)	Present several design architectures of software to the class	SMS 2.0 design architectures	Go Kiat Casey	Presented one of the architectures to class, and answered questions from class Presented the same design with Go to class, and took down opinions from class. Designed another two interface architectures for the software	2 2 2
7/09/03 7/10/03 7/11/03 7/12/03 7/13/03	Continue to implement new version of software	Finalized SMS 2.0	Casey, Go, Kiat		N/A
7/14/03	Finalize software & work on Final report	Finalized SMS 2.0 and Rough Draft of the Final Report	Casey, Go, Kiat		N/A
7/16/03	Final team presentation	Presentation to class	Casey, Go, Kiat		N/A
7/21/03	Assemble final report	Hardcopy of team presentation	Casey, Go, Kiat		N/A
7/23/03	Submit final report	Final report	Casey, Go, Kiat		N/A
7/24/03	Final presentation	Final presentation & poster	Casey, Go, Kiat		N/A

Bibliography and References

Bibliography:

- www.experts-exchange.com
- www.google.com
- www.asp.net
- www.webwizguide.com/asp/

*these links provide accesses to code examples and syntax references that aid our module in developing and implementing the SMS software.

- http://www.epistemelinks.com/Info/OpSys.aspx
- http://www.linuxfocus.org/English/May1998/article41.html
- http://www.chilisoft.com
- http://us4.php.net/manual/en/
- http://www.sun.com/smi/Press/sunflash/2002-03/sunflash.20020326.2.html
- http://sqlcourse.com/intro.html

*these links provide us information when we were doing the research for this midterm report.

Course Reading and References:

- Spring 2003 IPRO327 Final Report
- Spring 2003 Project Plan
- Spring 2003 Midterm Report
- Summer 2003 IPRO327 Project Plan version 2
- Family Business Autumn 2002
- *Using the Balanced Scorecard as a Strategic Management System* by Robert S. Kaplan and David P. Norton, Harvard Business Review January-February 1996.
- *Beyond the numbers* by Dylan Miyake, Intelligent Enterprise July 26th 2002.

Contact Information

Casey Ligas
ligacas@iit.edu
[redacted]

Go Nakagawa
nakaggo@iit.edu
[redacted]

HuanKiat Tang
t_huankiat@iit.edu
[redacted]

Signatures

_____	_____	_____
Casey Ligas	Go Nakagawa	HuanKiat Tang