BUSINESS SIMULATIONS: FROM PUNCH CARDS TO WEB-BASED

Kenneth R. Goosen
Micro Business Publications, Inc.
krgoosen@cei.net

ABSTRACT

The purpose of this paper is to present an overview of the components including various Web pages necessary for constructing a Web-based simulation. As well be seen, there is considerable complexity in designing and making operational a Web site capable of processing a business simulation. The major components and elements of a Web-based business simulation are outlined and explained.

INTRODUCTION

The amazing development of the World Wide Web since 1995 and the advancement of highly sophisticated Web browsers has created a technology rich in educational opportunities and potential for learning in many fields of learning including business. In recent years, a number of existing and new business simulations have become totally Web-based. This fact has been recognized in various papers presented at ABSEL, but there have been no papers which have addressed how to construct a Web-based simulation including the server side programming necessary to run the simulation. The purpose of this paper is to present an overview of the components including various Web pages necessary for constructing a Web-based simulation. As well be seen, there is considerable complexity in designing and making operational a Web site capable of processing a business simulation.

BACKGROUND

In many cases, business simulations have a long life, some over 40 years. In order to have a long life, simulations must evolve with changes in computer technology. My own simulation, *The Management/ Accounting Simulation*, which was first developed in 1969, consisted of well over 1,000 punched cards. It required a box 8” by 15" by 3 1/2“and in order to process the program the boxed set of cards had to be taken to computer center and dropped off. If lucky, results might be obtained the next day. My simulation is now in a 4th edition and to get to this state the simulation had to evolve through a rigorous computer programming path. Every few years substantial new programming had to take place.

The initial programming language used was FORTRAN. Around 1980, with the introduction of personal computers, I reprogrammed the simulation in BASIC. At that time the dominant language of personal computers was BASIC. In 1988, Microsoft developed and released QuickBasic. Converting from BASIC to QuickBasic was a challenge and required considerable effort. But the results were some graphic features and other features not previously available.

In the early years of the 1980s, floppy disks were the dominant form of input and the means for saving files independent of a hard drive. Early in the 1990s, the 3.5 inch disk replaced floppy disks which became a convenient means for students to record their team decisions. The amount of memory on a 3.5 disk was substantially greater. Then the compact disk (CD) soon replaced the 3.5 disc. Computers now no longer have 3.5 drives; they have been in part replaced by Flash drives (thumb drives) which easily insert into a USB port.

The development of personal computers in the late 1970s and early 1980s and substantial advances in software and hardware made possible major improvement in processing of simulations and also made use and distribution of simulations much easier. No longer did simulation programs require the distribution of bulky boxes of punched cards. Starting round 2000, it was fairly easily for users of simulations to download simulation software from a Web site. Now Web-based simulations no longer require the user to download and install software.

Initially, the operating systems of personal computers were initially DOS based. In order to develop an effective menu driven program, considerable programming was required and then the end results did not come close to the slick appearance and effectiveness of Windows based-programs that were possible in the late 1990s and early 2000 years. The impact of the evolution of operating systems from DOS to Windows 3.1 to Windows 95 to Window 1998 to Windows XP to Vista and finally to Windows 7 proved to be a major challenge to simulation developers. The problem with these changes every few years was that substantial reprogramming had to take place. For example, programs written in VB5.0 were fully supported by Windows 98; however, the Windows XP operating system did not fully support programs written in VB5.0. Those simulations which were not reprogrammed in VB6.0 or some equivalent windows program immediately became obsolete. My own simulation, *The Management/Accounting Simulation*, and Bernie Keys simulation, *MMG,* and Ron Jensen’s *BML* simulation could not be run using Windows XP without additional programming. In 2000, I reprogrammed my simulation using Visual Basic 6.0. *MMG* and *BML* were not converted until 5 years later and in the mean time all adoptions had ended. The time delay was too long to revive those two simulations. Long term survival of a simulation required immediate adjustment and reprogramming as changes occur in operating systems. The evolution of operating systems from DOS to Basic to QuickBasic to Visual Basic for DOS and then to Visual

Basic for Windows has been traumatic at times for authors of business simulations.

NEED FOR WEB-BASED SIMULATIONS

In 1998, it was apparent that in order for my simulation to survive, I had to convert from a DOS based program to a Windows based program. Hiring programmers to make the conversion was cost prohibitive, so I began a project of teaching myself the fundamentals of Visual Basic 6.0, a marvelous and in part an intuitive method of programming. After about three months of intensive study and experimenting with the programming language, I began the task of converting my simulation from QuickBasic to VB 6.0. To my surprise, I had completed the task about six months later. The object oriented nature of VB 6.0 and the numerous controls such as buttons and text boxes made fairly easy the creation of a nice looking interface. The various screens simply required a mouse click to go from one page or screen to another. The simulation was now attractively menu oriented and navigation from one activity to another was easy and quick.

Now another technology has made simulations written in programs such as VB 6.0 nearly obsolete. The tremendous advancement of Web-based software and powerful server-side programming languages has made it imperative that existing simulations or new simulations be Web- based. It is unlikely that business simulations that are not Web-based can compete against Web-based simulations.

Converting a simulation to a Web-based simulation requires certain skills:

1. Ability to program in language such as VB.net or C#.
2. Ability to create server-side programs using a language such as ASP.NET 4.
3. Ability to create a database either text file (flat style) based or a sophisticated relational database.
4. using a database program such as Microsoft Access.
5. Ability to write a concise but content rich student manual.
6. Ability to develop a complete mathematical model of a business and economic environment.

The model must be realistic and contain mathematical functions and algorithms that generate realistic results (e.g., financial statements) .

The skills required now may be beyond the capability of one person; consequently, a team approach of three or more individuals may be required.

WEB-BASED SIMULATION COMPONENTS

There are four main components of a Web-based simulation:

1. A web site (its own domain name is preferable)
2. Instructor's area
3. Student's area
4. Administrative area

Items two, three, and four require development of forms or pages that capture inputs such as students decisions and make output that consists of financial statements and other information. The initial or starting page (technically often called the default page) may contain general information about the simulation, but the main purpose is to allow registration of professors and student. The ability to log-in with a password is critical. Professors (instructors) will log-in to the instructor's section and students will log-in to a separate student section. Logging in by instructors will take them to a page that I call the instructor's page and log-in by students will take them to the main student's page.

The instructor's sections (various .aspx pages, assuming ASP.NET 4 is used) will consist of the following pages:

1. Instructors' main page
   The main element of this page will be a menu bar that allows the users to navigate to various necessary simulation activities such as:
   a. Enter decisions page
   b. Change parameters page
   c. Setup new industry page
   d. View/print financial statements page
   e. Compute results page
   f. Create team password page
   g. View contents of industry folders
2. The student section or part of the Web site has less options, but at a minimum consists of:
   a. A main student page containing a navigation menu bar.
   b. Enter decisions page
   c. View/print output (financial statements) page.
3. Administrative Section

The Web site must contain several features that are reserved strictly to the Web-site master and not available or accessible to professors or students. This part allows the administrator to :

a. Create a special folder for each individual professor user of the simulation.
b. Create industry subfolders within each professor folder.
c. Create a password file containing all of the passwords of registered users.

The following diagram summarizes the various parts of a simulation-oriented Web site: (Please See Figure 1)

DATABASE DEVELOPMENT

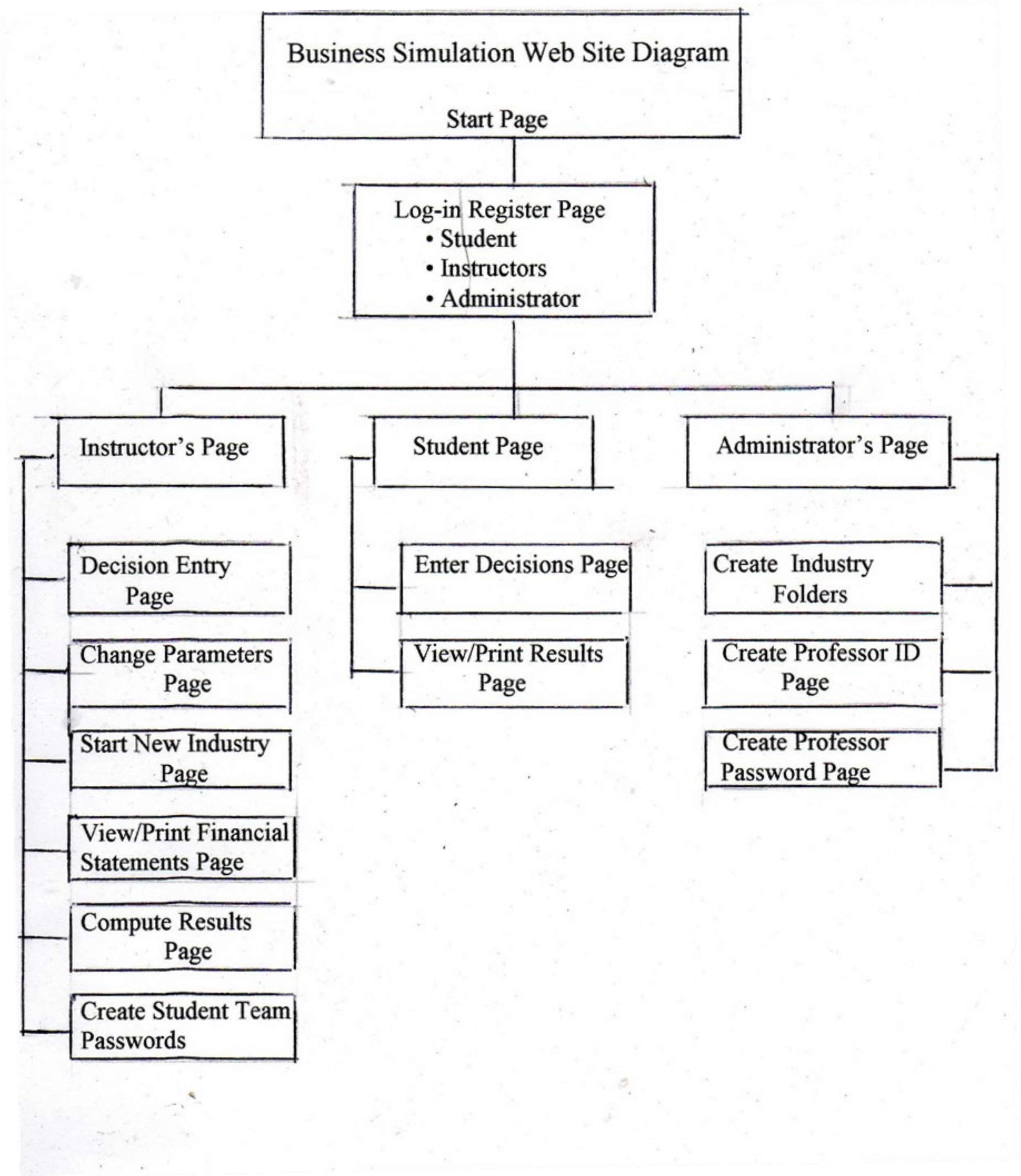
e Web-based simulation I developed uses a folder/text file type database. In one sense, it is not a true database. However, unless the number of professor users exceeds say several hundred this type of data works very well and can be very efficient and effective. In my simulation, and I assume in all simulations of various types, it is necessary to capture or make available the following kinds of data:

1. Decisions
2. Parameters (which may be changed by the user)
3. History data
4. Professor IDs (user names)
5. Professor passwords
6. Team passwords
7. Starting history

In my case, the above data will reside in various text files in various folders. In my simulation, these types of data contain the following names:

- Decision files (DT1P1IInd1.txt , where T1 represents team number, P1 represents period number, and Ind1 is the name of industry 1)

Figure 1
Parts of a Simulation-oriented Web site



- Parameter file (Parameters.txt)
  - History data (HT1P1IInd1.txt)
- Financial statements (FST1P1IInd1.txt)
  - Performance evaluation (PEind1.txt)
  - Professor Ids (ProfID.txt)
  - Professor Passwords (ProfPW.txt)
  - Team passwords (TeamPWInd1.txt)

The main folder of the database is a folder which, for no particular reason, I labeled PCfolder. The folder then contains separate professor ID subfolders. For example, assume we have three potential users whose names are Ken Goodwin, Mary Smith and Jack Brown. Then within the PColder would reside the following subfolders: Kgoodwin, Msmith, Jbrown. Also, within each professor folder whose names are identical to the professor ID (user names) are industry subfolders. In my own case, I have limited these industry subfolders to four. I labeled each industry subfolder as Ind1, Ind2, Ind3, and Ind4. The number of industry folders can be greater than four if it is believed more than four are needed for any particular user.

The main folder of the database is a folder which, for no particular reason, I labeled PCfolder. The folder then contains separate professor ID subfolders. For example, assume we have three potential users whose names are Ken Goodwin, Mary Smith and Jack Brown. Then within the PColder would reside the following subfolders: Kgoodwin, Msmith, Jbrown. Also, within each professor folder whose names are identical to the professor ID (user names) are industry subfolders. In my own case, I have limited these industry subfolders to four. I labeled each industry subfolder as Ind1, Ind2, Ind3, and Ind4. The number of industry folders can be greater than four if it is believed more than four are needed for any particular user.

BASIC TEXT FILES REQUIRED IN THE SIMULATION

What ultimately resides in each industry subfolders are the following text files:

- Decision files of all team in the industry
- Team password files for all teams
- Financial statements files for all teams
- History files for all teams
- Performance evaluation file
- Parameters file for the industry (The parameters file may be different for each industry.)

For example, the path for saving a decision file would be, assuming the professor is Ken Goodwin and the team is team 1 and the period of play is period 1 and the industry is Ind1: ~/pCfolder/Kgoodwin/Ind1/DT1P1IInd1.txt

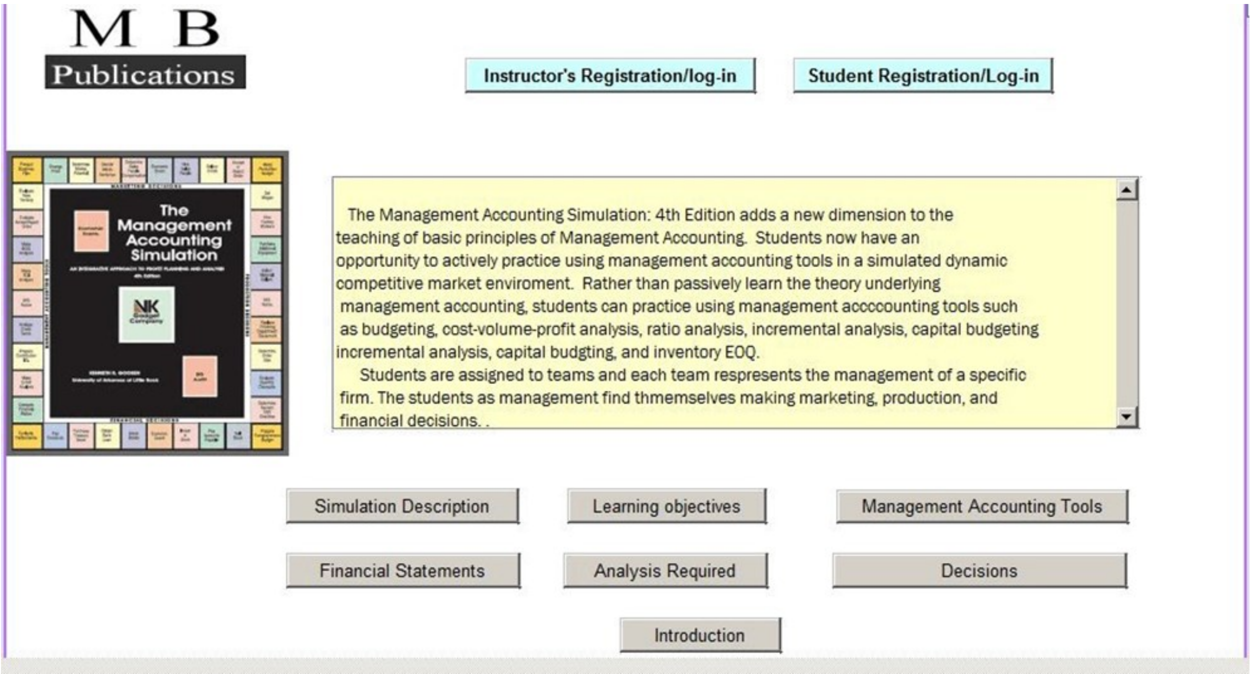
Because there would be in, fact, a number of professor users, it is necessary in the path to use variable names to identify professors, industries, periods and team numbers:

Let:

Tmn	- team number
PD	- period
Ind1	- Industry
ProfID	- Professor name

Then technically a more accurate description of the path then would be: ~/PCfolder/ProfId/Indu/DT + AT@ + Tmn + AP@ + PD + AI@ + Indu + A.txt@

Exhibit 1


M B
Publications
Instructor's Registration/log-in
Student Registration/Log-in
The Management Accounting Simulation
Simulation Description
Learning objectives
Management Accounting Tools
Financial Statements
Analysis Required
Decisions
Introduction

The Management Accounting Simulation: 4th Edition adds a new dimension to the teaching of basic principles of Management Accounting. Students now have an opportunity to actively practice using management accounting tools in a simulated dynamic competitive market enviroment. Rather than passively learn the theory underlying management accounting, students can practice using management acccounting tools such as budgeting, cost-volume-profit analysis, ratio analysis, incremental analysis, capital budgeting incremental analysis, capital budgting, and inventory EOQ.

Students are assigned to teams and each team respresents the management of a specific firm. The students as management find thmemselves making marketing, production, and financial decisions. .

In the PC folder would reside the following files:

- Start history
- Professsor ID file
- Professor Password file

The use of professor IDs, team passwords, and team numbers make the results of one team not visible to other teams. The only information all teams share in common is the performance evaluation file.

The ASP.NET 4 SERVER-SIDE PROGRAMMING LANGUAGE

The creation of a Web-based simulation requires the use of a server-side programming language. The language I used was ASP.NET 4. Each form or page consists of two files having the extension .aspx and .aspx.vb. Assume the form in question then is Form 1. Technically, the file name for the two files would be: Webform1.aspx and Webform1.aspx.vb. The Webform1.aspx, for example, could be renamed. One or more Web .apsx forms would contain the mathematical model of the simulation. The programming code used in a form such as Webform1.aspx would either be Visual Basic or C#. In my case, because of my previous experience with Visual Basic, I used Visual Basic 2010 Express.

The ASP. NET 4 language toolbox contains many controls, far more than I found necessary to use. The main controls I found the most useful were:

- Buttons
- Textbox
- Dropdownlist
- Option button
- List box
- Table

The development of a Web-based simulation requires that the author or one of the authors have considerable programming skills in either Visual Basic or C#. Also, the development of a Web-based simulation or the conversion of an existing simulation to a Web-based simulation will require a substantial devotion of time in two areas. First, the coding and development of the simulation requires considerable time. The process of testing and debugging may well require an equal amount of time. Second, The development of the required Web Pages for the Web-based simulation and the ability to navigate from one page to another will also be a time consuming activity.

It is, I believe, important to mention that the software I used was free. Microsoft makes free of charge Visual Studio Express. This software may be downloaded free by going to: http://www.microsoft.com/visualstudio/eng/products/visual-studio-2010-express.

The software, even though an Express version, is quite massive in size and contains several programs which were unnecessary for me to use. The two programs that I used were Visual Basic 2010.NET and ASP.NET 4. Visual Studio Express for me at first was rather intimidating. However, I found that the learning curve was not as steep as I first thought because of my prior knowledge of Visual Basic 6.0.

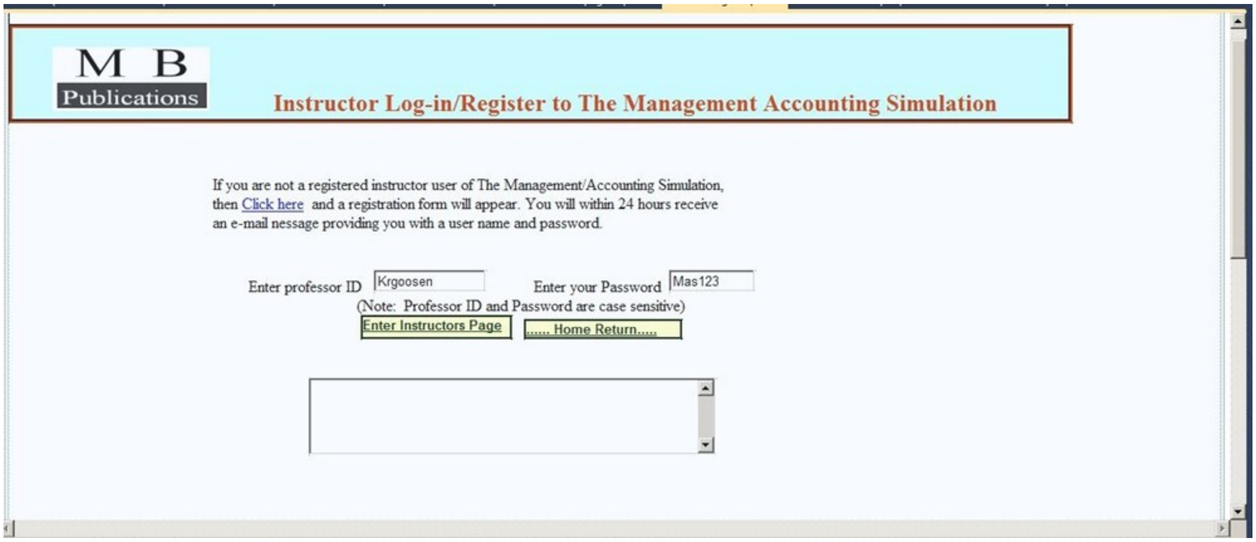
DESCRIPTION OF THE BASIC ELEMENTS OF THE MANAGEMENT/ ACCOUNTING SIMULATION

Navigation, that is, allowing the signed-on user to move from one page to another, requires development of a logical organizational structure of Web forms. Concerning the simulation I developed, part of the organizational structure is shown in Exhibits 1,2, and 3.

The first page that appears when a user enters the simulation Web site is shown in Exhibit 1.

This screen makes available some general information about the simulation. Also, two options appear for the next screen or page. Students will click on the Student Registration/log-in button and professors or instructors will

Exhibit 2



Instructor Log-in/Register to The Management Accounting Simulation

If you are not a registered instructor user of The Management/Accounting Simulation, then Click here and a registration form will appear. You will within 24 hours receive an e-mail nessage providing you with a user name and password.

Enter professor ID Krgoosen Enter your Password Mas123

(Note: Professor ID and Password are case sensitive) click on the Instructor's Registration/log-in button. When an individual instructor clicks on the log-in button, he or she will be taken to the screen shown in Exhibit 2. In order to arrive at the main instructor screen or page, the user will have to enter a professor ID (user name) and a password. If the correct user name and password are entered, then the user will be taken to the screen shown in Exhibit 3. In Exhibit 3, there is a menu bar (shown in blue) that provides six choices:

1. Compute Results
2. View/Change Decisions
3. View Financial Statements
4. Create/ Change Parameters
5. Create Team ID file
6. Home

It is somewhat apparent that a mouse click on one of the six choices will take the user to a separate page. The destination pages of the menu bar required extensive programming and, in my case, required programming in server-side Visual Basic. It is beyond the scope of this paper to present any of the code required. It addition to the menu bar, there are three other options in the form of buttons to click:

1. Set-up New Industry
2. View Decision Folder
3. View Instructions

These pages may be accessed for use only by the professor user. They cannot be accessed by student users of the simulation.

When a student successfully enters the professor name of his instructor and the assigned password of his team, the user is taken to the main student page. The student is then presented five options:

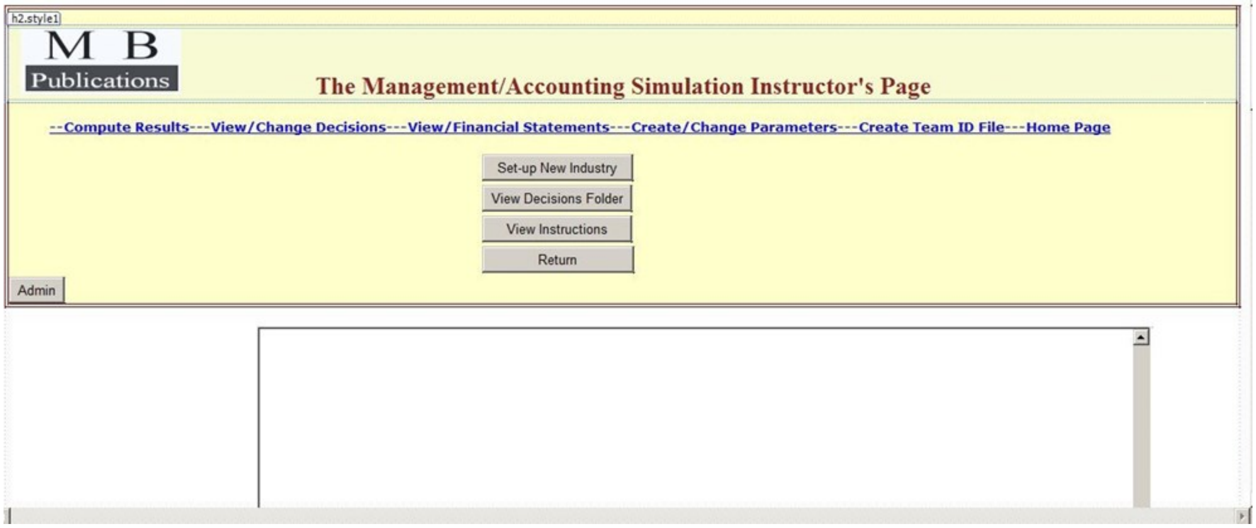
1. Enter Decisions
2. View Financial Statements
3. Print Financial Statements
4. View Performance Evaluation
5. Performance Evaluation

In order for an instructor user to use the simulation with a specific class, several housekeeping chores are required: (1) it is necessary select an industry name; (2) within that selected industry, it is necessary to transfer to the industry folder two beginning parameter files; (3) assign passwords to each team within the industry and create a password file. After these basic requirements are completed, students may go online to enter decisions and the user/instructor may process decisions

The development of a new simulation today compared to 40 years ago (when ABSEL held its first meeting) is considerably more difficult. In the current computer environment, the simulation developer must learn two programming languages rather than one. In my case, I had to have some skill in Visual Basic and ASP.NET 4. In 1969, when I began developing ***The Management/ Accounting Simulation.*** I only needed some programming skill in FORTRAN. However, on the other hand, a Web-based simulation simplifies a number activities:

1. No longer is the distribution of software to each individual required.
2. No longer is it necessary for the user to install software.
3. In the event of a correction required in the program, it is no longer necessary to send to each user a corrected software package.
4. No longer is it necessary for students to turn-in disks or thumb drives which contain their decisions.
5. There is no longer any need to upload or download decisions or output results.
6. Student can print output easily and quickly from a browser.
7. Easy on-line access to the simulation from virtually any location.

Exhibit 3


h2.style1
M B
Publications
The Management/Accounting Simulation Instructor's Page
--Compute Results---View/Change Decisions---View/Financial Statements---Create/Change Parameters---Create Team ID File---Home Page
Set-up New Industry
View Decisions Folder
View Instructions
Return
Admin

SUMMARY

The evolution of computers and programming languages require that business simulation be constantly revised or otherwise the programming language of the simulation becomes obsolete. This paper has described the evolution of one specific simulation, *The Management/ Accounting Simulation*, from FORTRAN, BASIC, QuickBasic, Visual Basic and now to ASP.NET 4. Also, during this same time, input sources have evolved from tape, floppy disks, 3.5 disks, CDs, and thumb drives. Now the dominant media for using simulations is the Web. Successful simulations at the current time must be Web-based; that is; programmed to run server-side.

At the time ABSEL began in 1974, there were many simulations available which were created by college professors/instructors. The number of the simulations exceeded over 150 business simulation, according to Zuckerman and Horn (1973). In 1972, *Contemporary Games, Vol 1*, by Jean Belch listed 242 business simulations and games. At the current time, the number of written by professors/instructors and currently operational is quite small. However, a number of ABSEL members who have developed simulations have successfully converted their simulations to a total Web-based simulation. Those ABSEL members who have done this include:

1. Dave Jordan, (http://www.bussim.info/index.htm)
2. Steve Gold, (http://www.goldsimulations.com/new/econSims/author.htm)
3. Phil Anderson, (http://oaktreesim.com/?/Business_Simulation_Game_Support)
4. Tim Scott, (http://oaktreesim.com/?Page=/OakTreeSim_About_Us)
5. Joe Wolfe, (http://onlinegbg.com/)
6. Ernie Cadotte, (http://www.marketplace-simulation.com/index.php)
7. Jerald Smith, (http://www.interpretive.com/rd6/index.php?pg=al_authors&sid=2)
8. Thomas Pray,(http://www.webdecidesim.com

To write and develop a Web-based simulation now requires a high skill level in computer programming. Fortunately, ASP.net 4 makes this task fairly easy for those that those previously having programming skills in Visual Basic Net. Such skill can be easily hired professionally; however, the total cost at a minimum would be between $20,000 and $50,000. In my own case, I devoted over 500 hours of my time to the conversion of my simulation to a Web-based server-side program. Had I hired the programming and at a rate of $50 an hour, I easily would have been out-of-pocket by at least $25,000. To develop a Web-based simulation at a low cost will now requires knowing a server-side programming language or having a co-author with such knowledge and willing to not charge for programming time. Also, a tremendous amount of spare time would be required for those that also teach full time.

REFERENCES

Belch, Jean. Contemporary Games, Volume I, (Detroit: Gale Research Company, 1972).

Zuckerman, David W., and Robert E. Horn. The Guide to Simulation Games for Education and Training, (Lexington, Massachusetts: Information Resources, Inc., 1973).

Spaanjaars, Imar, ASP.NET 4 in C# and VB, (Indianapolis, IN, Wiley Publishing, Inc.,2010)

Shepard, George, ASP.NET 4: Step by Step, (Redmond, Washington , Microsoft Press,2010)