APPORVABLE PAGE FOR GRADUATE THESIS OR PROJECT

SUBMITTED IN PARTIAL FULFILLMENT OF REQUIREMENTS FOR
DEGREE OF MASTER OF SCIENCE IN COMPUTER SCIENCE AT
CALIFORNIA STATE UNIVERSITY, LOS ANGELES

Andre Liv
Candidate

Computer Science
Field of Concentration

TITLE: Development of Web-based SimEcon Instructional Software

APPROVED:

Chengyu Sun
Thesis Chair
Signature

Jiang Guo
Faculty Member
Signature

Raj S. Pamula
Department Chair
Signature

DATE: 6-8-2007
DEVELOPMENT OF WEB-BASED SIMECON INSTRUCTIONAL SOFTWARE

A Thesis

Presented to

the Faculty of the Department of

Computer Science

California State University, Los Angeles

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

by

Andre Liv

June 2007
Acknowledgments

I would like to thank the Lord for His unconditional love, and to Dr. Sun for advising and guiding me in this project. I'm grateful to every faculty in the CS department for being knowledgeable and Tricia for being so helpful and kind. I would like to give thanks also to my parents for supporting me financially and my beloved girlfriend for her motivation and support.
Abstract

Development of Web-based SimEcon

Instructional Software

By

Andre Liv

In the past decade, the World Wide Web has enabled the fast delivery of applications to users and levered the limitations of application accessibility. Hence it is becoming imperative to develop web-based solutions to address the growing need of getting the software into the general public with ease of use and fast access. As part of this growing trend, converting SimEcon software into a web-based format will improve the chances that students will use the application and increase the likelihood of instructors on this campus and others using this software in their economic courses, also it will greatly increase the likelihood that it will be picked up by a commercial publisher.

Economics is a challenging field of study and often times business students hesitate to enroll in Economics courses if they are not required for their major. Consequently, fewer and fewer Economics courses are being offered by some colleges or universities because of the lack of students' interest in Economics. Dr. Garston (CSULA) and Dr. Bresnock (CSUP and UCLA) developed SimEcon software to help students develop and improve their learning in Economics studies. SimEcon is a simulation software designed to be used as instructional tool. The software consists of sixteen
standalone programs called *module*. These *modules* are Markets, Competition, Monopoly, Labor, Macro1, DrugWars, Banking, E-Growth, UtilityMax, CostMin, GasStations, Externalities, Macro2, Macro3, Exchange and Distribution.

SimEcon is shown and proven to be useful and effective but it's a desktop application which requires that each module be individually downloaded and installed first in the computing labs or on students’ home computers before they could be used. The solution is to develop a Web application to address this issue, thus the goal of this project is to provide a web-based version deployed on Microsoft's ASP.NET 2.0 to be delivered for the World Wide Web to improve accessibility, save students' time and effort from installing the programs and simplify the maintenance and upgrade of the software. For this project, eight modules were completed and made available on the web for students to use. The web-based version will use the following technologies: ASP.NET 2.0 written with Visual Basic, Master Pages, XHTML, CSS.

This project is made possible with the first round of funding by the NSF (National Science Foundation) and it will be used at the basis for developing the remaining conversions if properly funded by publisher's contract. For this project, there are eight modules to be converted which include the following modules: Markets, Competition, Monopoly, Labor, DrugWars, E-Growth, UtilityMax and Distribution. Currently we have completed the conversion of the above mentioned modules which are currently used in Economics courses.
Table of Contents

Acknowledgments ........................................................................................................ iii

Abstract ....................................................................................................................... iv

List of Figures ................................................................................................................ iv

1. Introduction ................................................................................................................ 1

2. Technological Background .................................................................................... 3

3. SimEcon Modules ................................................................................................... 8
   3.1 Markets Module ................................................................................................ 8
   3.2 Competition Module ....................................................................................... 9
   3.3 Monopoly Module .......................................................................................... 10
   3.4 Labor Module .................................................................................................. 11
   3.5 UtilityMax Module ....................................................................................... 12
   3.6 DrugWars Module ......................................................................................... 13
   3.7 E-Growth Module .......................................................................................... 14
   3.8 Distribution Module ....................................................................................... 15

4. Design and Implementation .................................................................................... 17
   4.1 Design Issues .................................................................................................. 17
      4.1.1 Re-implement Page Controls ................................................................. 17
      4.1.2 Fine-tune the User Interface ................................................................. 18
      4.1.3 Port the Core Program Logic ................................................................. 18

vi
4.1.4 Browser's Back/Reload Button ................................. 19

4.2 Implementation Details .............................................. 20

4.2.1 Graphs Rendering .................................................. 21

4.2.2 Error and Warning Pages ...................................... 22

4.2.3 Back Button Behavior ............................................ 23

4.2.4 Provide Cross-browser Support ............................... 24

4.2.5 HTMLHelp Pages .................................................. 25

6. Conclusion ........................................................................ 26

Bibliography ........................................................................... 27
List of Figures

Figure

1. Initial Equilibrium ........................................... 1
2. HTTP Request/Response ...................................... 3
3. Master Pages .................................................. 6
4. Markets Module .............................................. 9
5. Competition Module ......................................... 11
6. Monopoly Module ............................................ 11
7. Labor Module .................................................. 12
8. UtilityMax ..................................................... 13
9. DrugWars Module ........................................... 14
10. E-Growth Module ........................................... 15
11. Distribution Module ....................................... 16
12. Error Message Web Page ................................... 19
13. AC and MC Curve .......................................... 21
14. Table Formatting ........................................... 24
1. Introduction

With the popularity of the Internet, more and more web applications have now replaced desktop applications. In recent years, we have witnessed an increasing number of educational institutions starting to provide e-learning tools to help students learn various subjects and courses such as Economics theories, and so on.

SimEcon was designed for “Principles of Economics” courses, meant to be used as a learning tool created to add value to Economics education, improve students learning, and attract students to Economics. It is a simulation software consisting of 16 modules. Each module contains an initial conditions page which displays the initial values before the users can input values and compare the results.

For instance in Markets Module, it starts out by displaying the initial conditions for demand variables and supply variables and initial equilibrium for price of wheat.

Figure 1: Initial Equilibrium
The users can then enter the market by trying to regulate the market or disturb the market price and quantity for wheat. In market regulation, the users can change the regulation for quantity or price and observe the results for production or price control. The system provides the users with hints when the price or quantity entered is too low or too high, and the users can adjust or vary the input to see and compare the results. In market disturbance, the system provides initial values for demand and supply variables for which the users can change the input values to increase or decrease the demand and supply for wheat which affect the quantity or price of wheat. In Markets module, the users learn to disturb the market for wheat by changing the initial value for price of corn, price of butter, income, rainfall, number of farms and seed cost which affect the equilibrium quantity and equilibrium price of wheat.

While the current SimEcon software is a desktop based, the web-based version will be deployed on the World Wide Web that will dramatically improve students' accessibility and takes away the burden and necessity to have students download and install the individual programs in order to be able to use them. Some students may not have the appropriate operating system or platform requirements to run the program which runs on Windows machines only. But to use the web version, most users just need to have Internet access and a web browser to use SimEcon.

The web-based version will make easy use of and access to the application, and eliminates any possible problems relating to download and installation issues. For
instance, most software can not be downloaded in most computing labs because of campus policies that prohibit students from downloading or installed software to lab computers. Likewise, installation of SimEcon software can be challenging in some cases because the software is platform dependent and can only be installed on Windows machines but not on Macintosh nor Linux machines. This can be really a problem for students who do not own a PC (Windows) at home, thus the web-based format can be of great benefit because the application can be used on any computers with Internet access and work on most browsers, such as Internet Explorer, Firefox, Netscape etc.

This project covers the conversion of eight modules which have been completed. The conversion uses ASP.NET with Visual Basic codes that are easy to maintain or change. However the implementation to web-based is challenging because of browser display issues, control functional behavior differences, input validation and browser's Back button etc that need to be considered and addressed.
2. Technological Background

A Web application is a software that provides services that can span over the World Wide Web. Users can access the Web application, which resides on the server, with a Web browser that sends HTTP requests to the Web server to request services. In return, the Web server uses HTTP responses to provide the service requested. There are two types of Web pages: static Web pages and dynamic Web pages. Static Web pages are Web pages written purely with HTML and generally display similar and static content.

As opposed to static Web pages, dynamic Web pages generate “on the fly” Web content that is created only when the page is requested. All operations are transparent to the client, the Web server processes the request and delivers dynamic content pages to the client through HTTP responses (see Figure 2). SimEcon is an example of dynamic Web applications written on ASP.NET with a combination of Visual Basic, XTHML, and CSS.

![Deliver Dynamic Content](image)

*Figure 2: HTTP Request/Response*
Web-based SimEcon is designed to work on Windows Server 2003 running Internet Information Services (IIS) 6.0 which provides a set of services that enable the development of scalable Web sites, applications and services. The benefits of deploying IIS 6.0 is to increase Web application reliability and availability. IIS 6.0 provides support for application developed in ASP.NET 2.0.
ASP.NET 2.0 allows developers to work in various languages that include VBScript, JavaScript, Java and C/C++ and so on. There are many web technologies such as J2EE, PHP, ColdFusion and ASP.NET that can be used to create responsive and dynamic web contents, however, for this project the conversion of SimEcon is done using ASP.NET 2.0. ASP.NET 2.0 is developed by Microsoft as the improvement over ASP.NET 1.0 and 1.1 because of their limitations. As the market for web development tools has become more competitive with the rising of PHP and JSP, and ColdFusion. ASP.NET 2.0's added new features and improvements allow web developers to increase productivity by creating content web pages with fewer codes. Among ASP.NET 2.0 newly improved features one of which is Page Framework also known as Master Pages.
With Master Pages, one can create a consistent “look and feel” across every web page by defining a template that can be reused on every Web page. Master Pages allows to define a specific region to place the header which contains individual Web page subtitle and footer used to contain the navigation controls such as buttons (see Figure 3). In the below Master Pages, we use a ContentPlaceHolder to define the page header to hold the page subtitle. Similarly, ContentPlaceHolder is used to defined the page footer that holds the page control buttons.
SimEcon Web pages also uses CSS to define the fonts, colors and layout throughout the pages. CSS presents a consistent page layout throughout the Web pages. It creates common font type, color and size for text labels on all pages as well as the button and text box size and location on all pages. More importantly, the use of CSS not only allows to create consistent visual display across the pages but also it ensures that the display looks coherent on most browsers such as Internet Explorer, Firefox and Netscape.
3. SimEcon Modules


For this project, eight modules were selected by Dr. Garston to be converted to web-based for the present time. The remaining modules will likely be converted as well if additional fundings are allocated to the project.

3.1 Markets Module

The Markets module (see Figure 4) contains two parts. The first part allows students to get familiar with the basics of supply and demand in a competitive market. The second part allows students to interfere with the market as government does which includes the imposition of price ceiling, price floor and output regulation and observe the consequences of the interference. In this module, students can decide to regulate the market or disturb the market for demand by changing the price for butter, corn and income. Students can also disturb the market for supply by changing the numbers of farms, price of seeds and rainfall and compare the results and observe the shifts in
demand and supply. After disturbing the market, students again can have the option to regulate the market and see the results which is compared to the new equilibrium.

**3.2 Competition Module**

The Competition module (see Figure 5) allows students to practice making profit maximizing decisions for a firm in a competitive market. Students can either select different market prices or let the system generates different market prices for the firm. The firm then tries to maximize the profits or minimize losses for a given market price which can be affected by costs and production changes. The module begins by showing information on short run costs at selected output levels. Students can then guess or approximate the profit output and see the marginal cost that is resulted and compare the marginal cost (MC) to price and the average cost (AVC) to price.
3.3 Monopoly Model

The Monopoly module (see Figure 6) allows students to learn how to achieve profit maximizing output for a monopoly firm and learn how the societies use the three regulation levels (marginal cost based regulation, average cost based regulation, and anti-trust policy) to control the monopoly firms. Students set the entry cost and output levels for the firm then apply the three levels of regulation and observe the consequences for the firm and consumers to observe the results for revenues, costs and profits. The module allows students to determine the level for fixed cost for a firm and compare the results for price, marginal revenue, marginal, average, total cost etc and see the output in a graph. Students can then regulate the market by applying marginal cost price regulation, average cost price regulation and anti-trust regulation to compare profit for the firm.
3.4 Labor Module

The Labor module (see Figure 7) has to parts. The first part shows the market for bakers. The second part allows students to learn how the degree of disturbances can affect the labor market participants in terms of wages and employment levels. It shows that some types of anti-discrimination policy work and some don't. Also there a two ethnic group in society, the Big Pildians and the Little Pildians. Big Pildians often discriminate against Little Pildians. The module begins by displaying the initial conditions in the labor market, students can choose either to disturb the market or discriminate the labor market.

To disturb the market, students can change the price for bread, cost of oven, price of flour, population, participation rate and the wage, and then compare the results. Also, students can set the level of discrimination against a disfavored group that affect workers'
wage and employment.

**Figure 7: Labor Module**

### 3.5 UtilityMax Module

The UtilityMax module (see Figure 8) allows students to learn about how consumers can maximize utility based on limited income. Students can change the demand for two goods (Beer and music CDs) depending on consumer’s preferences based on prices and income, and observe the results. An increase in price of a good affects the amount consumers will buy that good, and price increase for one good can affect the amount of another good a consumer buys, and this is called the “Law of Demand”. The module begins by displaying the initial budget information that includes the consumer’s initial income and initial price for goods (Beer and CDs). Students can then decide to set utility function parameters or let the system to generate those parameters. There are two parameters, the first is the relative weights of the goods and the
second is the degree of substitutability. Then students must choose a combination of Beer and CDs by setting the values on or off the budget line and see the results in numeric form or graphical presentation.

![UtilityMax Module](image)

**Figure 8: UtilityMax Module**

### 3.6 DrugWars Module

The DrugWars module (see Figure 9) allows students to understand the uses of cost-benefit analysis and its limitations. Students are given a limited budget to fight the war on drugs through various programs that include therapy, law enforcement, and education. The objective of this module is to make the best use of the budget and apply the equimarginal principle to maximize the total net benefit to society. This module begins by showing the damages done to property, drug-user lives, and non-user lives by drugs if no money is given to fight against drugs use. Students' task is to set the dollar values of lives for users and non-users and allocate the $100 billion budget on therapy,
law enforcement, and education to fight and to minimize the damages of drugs use.

Students then can see the results that display the public approval rating and marginal results. The public approval rating indicates the public satisfaction of students' performance in fighting drugs use. The marginal results tell students if the allocation of the budget satisfies the equimarginal principle.

![Drug Wars Module](image)

*Figure 9: DrugWars Module*

### 3.7 E-Growth Module

The E-Growth module allows students to get an understanding of economic growth and development. Students are given the responsibility to manage and increase the nation's economy growth by manipulating the social and economic determinants consisting of age of marriage, number of children and R&D spending. Students determine how each variable affects growth in real GDP and real GDP per capita by
comparing the results (generation by generation) for six consecutive generations as the decision can create large and rich societies, and small rich or large poor societies. This module begins by providing the initial information about the state of the nation in Generation 0, students then enter age of marriage, number of children and R&D spending, amount of investment and public goods for each generation and observe the nation's economic growth.

3.8 Distribution Module

The Distribution module (see Figure 11) allows students to examine how various policies and disturbances affect the distribution of income which can be used as an indication of how good a society is. The distribution of income differs among societies in the USA where the distribution can be affected by a number of variables such as government policies and no-governmental disturbances. The module provides data on
“current” distribution of income and students can choose to adjust the distribution of income by changing the disturbance or government policies and compare the results in numerical format or using Lorenzo Curve.

\[\text{Figure 11: Distribution Module}\]
4. Design and Implementation

This section presents the design issues and implementation details of the project. Like most Web applications, it is sometimes challenging to duplicate the functionality, behavior and graphical “look and feel” layout of the original desktop version. SimEcon was not developed for the purpose to be deployed on the World Wide Web, however the web-based version is designed to work as closely as possible to the desktop version in terms of functionality and design layout.

4.1 Design Issues

The conversion of SimEcon modules to web-based format takes into account the following considerations: port the core program logic, simulate the navigational behavior, redesign the user interface, and support cross-browser display.

4.1.1 Port the Core Program Logic

With the original codes converted to ASP.NET 2.0 using VisualBasic.NET, it is necessarily to port the core program logic. With ASP.NET 2.0, some portions of the original codes can be reused but for the most part, it needs to re-implement the remaining code in VB which can affect the way the logic of the program is written.
4.1.2 Simulate the Navigational Behavior

SimEcon software uses Windows controls to allow users to perform various tasks such as navigating between pages, returning to previous pages or validating users' input. These controls are converted to Web controls. For instance, there are 'Back' buttons that take two pages back, and this can be hard to implement using the Internet browser's back button because it only allows the user to return to previous page, only one page at a time. Also, the browser's 'Back' button cause sometimes the Web page to reload twice which causes input validation issues. To resolve this issue it is necessary to add a back button within the Web page to perform the Windows 'Back' button functionality and disabled the browser's 'Back' button.

4.1.3 Redesign the User Interface

SimEcon software uses pop up windows to display error and warning messages. To implement the same functionality in web-based, it is necessarily to use JavaScript however because of some browser settings which block pop ups, it can be difficult to recreate similar behavior. To address this issue, warning or error messages are displayed on a Web page (see Figure 12) instead of a pop up window.
4.1.4 Support Cross-browser Display

The adaption of the user interface into the web-based version can be challenging because some Web pages do not always display consistently when viewed in different browsers. For instance, there are Web pages where text boxes or buttons sometimes look offset when viewed in Firefox but not in Internet Explorer. Thus it is important to review the page layout and look for inconsistency, and fixed them accordingly using Cascading Style Sheet with the position for text boxes, text labels and buttons set to 'absolute'.

4.2 Implementation Details

This project uses state-of-the-art technologies to create or render graphs, handle
error and warning errors, program the 'Back' button behavior, CSS to provide cross-browser support, XHTML to display help pages, and table formatting. It was developed in Visual.NET and ASP.NET using Visual Studio 2005, and the deployment in environment requires IIS 6.0 with ASP.NET 2.0 support.

4.2.1 Graphs rendering

The rendering code of the original code hardcodes the coordinates, but the rendering code of the web-based version adjusts the drawing of graphs based on the width and height of the drawing area. The rendering is formula-based in the sense that if the formula changes, the curves will change accordingly. To draw curves, it requires to draw multiple short lines. For instance to draw the 'AC' curve, the rendering code uses loops to draw lines that increment or decrement based of the formula used. Multiple lines are drawn for the loops and these lines are connected to each other to form the curve (see Figure 12). The curve needs to be scaled down to fit the drawing area/ It uses the extreme values to calculate the minimum and maximum values so that it will not exceed to drawing area.
4.2.2 Error and Warning Pages

The system uses error messages to tell the users that the input is incorrect. In some cases, the messages can be as simple as "the value is too low" or "the value is too high" but in most cases, the messages provide very specific information that explain the reason why the input is invalid. The implementation for handling errors can be done with various methods. For instance, each module can be designed to handle its own error messages but this implementation is not efficient and is not easy to maintain because that would require to create multiple Web pages to handle each and individual error message. The easiest way to implement this is to create a single and common interface to handle all the error messages for all the eight converted modules. Each time the users enter an invalid input, the page redirects the users to the error page which displays the error.
messages in question. The error page uses two session variables, one common session variable stores the error message, and another common session variable stores the page URI. This URI is used by error page to redirect the user back to the user page so that the users can input new values.

To handle the various warning messages, the implementation follows the same design for handling error messages. A single warning page is needed to display all the warning messages for seven modules. However, for E-Growth module, the implementation requires five separate warning pages because the implementation uses one session variable for storing the warning message and one for storing the page URI.

4.2.3 Program the 'Back' Button Behavior

The browser's 'Back' button allows to return to previous page, one page at a time, it can not duplicate the desktop 'Back' button functionality. Whenever the users click on it, it causes the page to go back or reload and this can cause a validation problem because the system uses a server-side method to check the users' input. The browser 'Back' button needs to be 'disabled' and an alternative solution is to add a 'Back' button to the Web page to allows the users to return to the intended Web page depending on the program workflow. The code uses the 'redirect' method to connect to the specified Web page which allows the users to go back to previous Web pages to enter new input.

4.2.4 Provide Cross-browser Support

The design adds a CSS file to provide cross-browser support. It defines the
attributes for class items and format the different elements such as text labels, button
controls, text boxes’ color, size, and more importantly the location of objects within the
pages. Some elements look fine on different browsers but some other ones (even with
position set to 'absolute') don't display perfectly consistent. This may be due to the
browsers' use of the coordinate system which varies from browsers to browsers. For
instance, Firefox 1.5 tends to display the page content a little low whereas IE 6.0 displays
the page content in a more consistent way.

4.2.5 HTML Help Pages

Every SimEcon module has its own help folder that contains all help files for easy
of maintenance. Help files are written with XHTML which makes it easy format text files
and insert diagrams and so on. Each help page uses dark blue background color while the
foreground text uses white color. To insert the diagrams which have a white background
and black foreground and text color, these diagrams need to be fine-tune using a graphic
editor such as Photoshop to remove the background color and change the foreground
color to white color.

4.2.6 Table Formatting

Each module displays the initial values and results in a table which formats the
values in a well organized manner. It displays the values in columns where text values are
aligned with left indentation and numeric values (dollar values) are aligned with right indentation. The example below (see Figure 13) has two tables. The table on the top uses 'Colspan' to create five columns. The table at the bottom uses 'Colspan' to create 4 column however the last column has no initial entry and the dollar sign is indented to the right.

Figure 14: Table Formatting
5. Conclusion

This project represents the ongoing goal of making software application readily to be used. It combines state-of-the-art technologies that provide tools to design and develop Web application to meet users' need and improve accessibility by getting the application to the general public with ease of use.

It's the achievement of a successful project on many levels. First of all, it allows the collaboration with Dr. Garston and Dr. Bresnock work had worked on and used the software in their classrooms to better students' learning and help them understand the Principles of Economics. It adds important values that contributes to the understanding of technologies for implementing the conversion of SimEcon to web-based format. This project also provides a deeper understanding of how a conversion process work which will be used a the basis for converting the remaining modules. And indeed Dr. Garston and Dr. Bresnock have proposed that the entire modules will be converted a well funded under publisher's contract.
Bibliography

http://www.w3.org/Style/CSS/.

[2] Poznan, Tomek, Client-server, 3 June 2007

