Proceedings of the 16th Annual NACCQ, Palmerston North New Zealand July, 2003 (eds) Mann, S. and Williamson, A. www.naccq.ac.nz

Moving from Classic ASP to ASP.NET

Dr. Xiaosong Li

School of Computing and Information Technology
UNITEC Institute of Technology
Auckland, NZ

xli@unitec.ac.nz

ABSTRACT

ASP.NET is Microsoft new offering for Web application development, innovation within ASP.NET have resulted in significant industry popularity for this product. Consequently there is an increased need for ASP.NET education. The Web Application Development is a third year undergraduate course. To meet the demands of both industry and students, we have changed the focus of this course from Classic ASP to ASP.NET. This paper reports this move. The significant features of ASP.NET and the motivations for this move are discussed. The process, the problems encountered, and some helpful online learning resources are described.

Key words

Web Application Development, Classic ASP, ASP.NET, Move, VB.NET

1. INTRODUCTION

ASP.NET is not just a new version of ASP. It provides innovation for moving Windows applications to Web applications. Web services and the .NET framework have made the vision of the Web as the next generation computing platform a reality. With server controls, Web forms and "code-behind", we can develop a Web application by using a complete object-oriented programming (OOP) model. This increases the popularity of ASP.NET in industry. The industry project is the final course of the Bachelor of Computing Systems (BCS) degree at UNITEC, in which students undertake a real-world project. We have observed a rapid growth of ASP.NET related industry projects in our school.

The Web Application Development (WAD) paper is a third year undergraduate course. It was originally offered using ASP 2.0 and ColdFusion. To meet the demands from both industry and students, we have changed the course content to cover ASP.NET, Visual Studio.NET (VS.NET) and ColdFusion. This change commenced with the first semester of 2003.

This paper will examine the features of ASP.NET and explain why these are unique. The motivations for moving to ASP.NET are discussed by analyzing the current situation of ASP.NET related to industry projects in our school, analyzing the results of short surveys on students, and analyzing whether ASP.NET is a better tool for teaching. Problems encountered during the move are also discussed and some of the learning resources are presented. It is anticipated that these will be helpful for teachers who intend to introduce ASP.NET.

2. WHAT MAKES ASP.NET SPECIAL?

There are many articles on the Internet discussing the advantages of ASP.NET over Classic Active Server Pages (ASP), such as that ASP.NET introduces an integrated development environment (IDE), a single

Table 1. The ASP.NET projects are increasing rapidly

Year/Semester	Projects Checked	ASP/ASP.NET Projects	ASP Projects	ASP.NET Projects
2002/Semester 1	17	6 (6/17 = 35%)	6 (6/17 = 35%)	0 (0/17 = 0%)
2002/Semester 2	17	7 (7/17 = 41%)	5 (5/17 = 29%)	2 (2/17 = 12%)
2003/Semester 1	19	8 (8/19 = 42%)	3 (3/19 = 16%)	5 (5/19 = 26%)

development library for all types of applications, compiled as well as strongly typed code, and a true OO approach to Web application development (Goodyear, 2002, Bloom, 2002).

Traditionally, we have three versions of ASP (ASP 1.0, ASP 2.0 and ASP 3.0), which are called Classic ASP. Although each version provides certain new features to overcome the shortcomings of its predecessors, these versions of ASP follow the same working model and share many limitations. Their successor ASP.NET supports complete new working model while preserving the traditional working model and provides innovative techniques to overcome the limitations of Classic ASP.

2.1. Architecture

ASP.NET enhances and extends the Windows DNA (Windows Distributed interNet Application). The windows DNA specification is a methodology for building n-tier applications using Microsoft (DCOM/COM) technologies. Breaking applications into functional pieces and deploying these across a network is a strategy to make better use of organizational resources. This needs a well-planned architecture. In the past, usually it was the windows DNA. DCOM communication normally has problems with firewalls and proxy servers. This means Windows DNA usually only works well within an intranet, not on the Internet. DCOM/COM also need registry entries.

ASP.NET makes the process of creating and integrating Web Services easier, which can be used in a similar manner to the Windows DNA. Here DCOM/COM is no longer involved. HTTP (as channels), SOAP (as formatters) and XML are used for communication and data-transfer between distributed components. This overcomes the problem of communicating across the Internet and across corporate firewalls without resorting to proprietary solutions that require additional communications ports to be opened to external access. In addition, URI (uniform resource identifier) and UDDI (Universal Description Discovery and Integration) are used for remote components references instead of registry entries.

2.2. Development

ASP.NET integrates seamlessly with VS.NET IDE. VS.NET includes built-in support for creating and modifying content. This unifies the ASP/VB programming models for the developers. Instead of opening multiple IDEs (as with Classic ASP platform), developers can open a single IDE and do all their work from a clean, consistent interface. VS.NET is equipped with powerful debugging environment. This means that the powerful debugger for Windows applications is now available to debug Web applications as well.

ASP.NET enables programmers to take advantage of the OOP model, for example, code sharing. Under OOP model, one of the most common ways to achieve code sharing is inheritance, which is not available in Classic ASP. Since complete OO features are supported in ASP.NET, developers can transfer their OO design smoothly into ASP.NET code, enabling a software company to keep their Windows application development styles, with which they are familiar, in Web application development; and also they can convert their Windows applications into Web applications without major modifications.

ASP.NET's improved state maintenance features enable us to provide users with Web applications that are richer and faster than Classis ASP (Olges, 2002). ASP.NET supports advanced session state management. There are two major problems with session management in Classic ASP: session objects are stored in the Web server memory and session IDs are stored on the client computers as cookies. These prevent session management from being efficiently implemented. ASP.NET solves these problems in two ways: it provides a "cookieless" option for session objects so that a session ID can be passed via URL; it provides three different session modes (in process, state server, and SQL Server), so that a session object can either be stored on the Web server, a remote server or a database.

3. THE MOTIVATIONS FOR MOVING

3.1. The industry motivation

I've checked almost all the industry projects in our school for three semesters on whether they are WAD related, if yes, then what tools they have used. Table 1 shows a brief summary of the results.

For these three semesters, the total ASP/ASP.NET projects are increasing, but slowly. However the Classic ASP projects are dropping quickly and the ASP.NET projects are increasing rapidly (in the speed of more than 12% per semester). This gives us an idea that ASP.NET is preferred over Classic ASP in industry especially given that ASP.NET is only officially first released in 2002. Our student's feedbacks from their industry communication confirm this view. A huge number of articles on the Internet also support this view. This encourages us to drop Classic ASP and move to ASP.NET in our WAD course. Higher education has over years recognized that it is a service industry and has to revaluate their approach in the industry by placing greater emphasis on meeting the expectations and needs of their stakeholders (Nair, 2002).

3.2. The student motivation

The students demand ASP.NET. When students enroll in our WAD course, most of them are aiming to become a professional software developer. As a matter of fact, some of them already are software developers, or they were software developers and are seeking to return to the workplace. Techniques highly demanded in workplace are of great interest to them.

A short survey has been given to past students and current students respectively. For the past students, among the 11 responses, 100% students still want to learn ASP.NET; and if they are given choice, 82% students prefer to learn ASP.NET rather than Classic ASP, 18% students like to learn both. These answers are also supported by comments, such as "*I would prefer to know the technology that the industry requires me to work with*", "*I would like to work in future as a WAD professional and I think ASP.NET would be useful in this field.*" For the current students, among the 16 responses, 75% students prefer to learn ASP.NET rather than Classic ASP. However, 25% students answered no idea. This could be due to that they lack of knowledge of Classic ASP. This survey is done after 6 weeks of teaching.

3.3. The pedagogical motivation

Pedagogically speaking, a good tool for industry is not necessarily a good tool for teaching. Is ASP.NET a better tool for teaching than Classic ASP?

ASP.NET provides much richer language features than Classic ASP. We often have options to perform certain tasks. A key benefit of ASP.NET is that there exists a more gradual transition in programming models from simple to powerful, or from easy to difficult. Although ASP.NET supports OOP model, you don't have to program by using that model. A Web form without "code-behind" will work perfectly. An ASP.NET web page in complete Classic ASP model will still work. Although ASP.NET is integrated with VS.NET, we are not limited to use VS.NET. A notepad and a FTP client with a pre-created Web application directory also allow us to develop a reasonably large ASP.NET application. With ASP.NET, we can either develop a large distributed application with numbers of Web services and consumers, or develop a single simple Web application. Therefore, ASP.NET provides sufficient room for us to organize course materials at a suitable level for the students. The challenge for a lecturer is how to settle in at the right balance of power vs. simplicity, or at the right balance of difficulty vs. ease.

ASP.NET offers a more conventional approach to programming than does Classic ASP. It possesses all the features of a modern programming language. The Classic ASP programming style favors developers coming from HTML coding background, whereas ASP.NET is more suited to professional software developers. Given our entire WAD students have taken C/Delphi programming courses, and our aim is to output software professionals, ASP.NET is a better teaching tool for us. ASP.NET enhances the programming concepts the students learned from the previous courses and provides a good bridge to Advanced Distributed Computing and Advanced Object-Oriented Programming.

4. THE PROCESS

Our first step was to learn ASP.NET. After reading books and online tutorials, the next step is practical. We set an implementation server on the laptop in a stand-alone environment. The .NET Framework requires IIS 5 and the above; Windows 2000 or Windows XP professional will work with .NET. However, Windows XP home edition or Windows 98 won't work. On the client side, we can either use VS.NET or WebMatrix. Among these, only VS.NET costs money. The .NET Framework is included inside the VS.NET package. We also can download the .NET Framework from the

Internet. After the .NET Framework is installed, the ASP.NET QuickStart Tutorial is set up. It is also found on the Internet. This tutorial is a good starting point for experienced developers. It is claimed that the readers "should be fluent in HTML and general Web development terminology. …… should be familiar with the concepts behind interactive Web pages, including forms, scripts, and data access." More complicated examples can be found from Microsoft .NET Framework SDK Documentation or Microsoft Visual Studio .NET Documentation.

The second step was to test the teaching environment. A teaching server was set up for the Intranet on campus. It is configured for the client computers in the teaching lab. VS.NET is installed on the client computers. VS.NET provides two ways to access the Web server: FrontPage server extensions and File share. The FrontPage server extension is used on our teaching server. Programming testing has been done on all the major aspects of WAD. Except a few special ones, most of the problems occurred during the testing were minor problems which, after the communication with our Web technician, were resolved.

Teaching materials have been updated. The major changes have been made on the data interaction, form and controls, application/session management, and error handling. Given that ASP.NET has made XML very practical and the using of Web service much easier. A lecture on XML and Web service has been added. As a result, ColdFusion lectures are reduced. The assessment has been adjusted accordingly.

5. THE PROBLEMS

We have to admit that ASP.NET with VS.NET is a much more complicated client server environment than the Classic ASP environment. This complexity comes from the ASP.NET configuration system and the integration between the client computers and the Web server.

On ASP.NET server, each level of the application directory can have a configuration file. All these configuration files are optional except Machine.config. A developer has full control over those optional configuration files. Developers become more involved with the server settings via these files. One problem that happened to several students and myself on our home servers is the permission problem. We found our ASP.NET applications didn't have permission to write to database/XML files. Microsoft (2003) provides three solutions to this problem. The simplest one is to change the Machine.config file and set the userName attribute to SYSTEM in the <processModel> section.

We observed that VS.NET behave differently in a stand-alone environment, a single user client server environment, and a multiple user client server environment. A few problems don't occur in the first two environments occur frequently in the last environment. The major one is when we try to create a new project or open an existing project, we often get an error message, "*The user name or password you entered is incorrect, or you do not have authorization to permit this operation*", even if our user name and password are perfectly correct. This problem seems to be caused by FrontPage server extensions. Regularly cleaning VSWebCache partially solved the problem. This approach is confirmed by Kiely (2003).

Another problem is a debug problem. When we try to use Debug|Start or Debug|Start Without Debugging in the multiple user client server environment within VS.NET, we often get error messages. "*…… Unable to start debugging on the web server. ……"*. However, we don't have the same problem for Debug|Start Without Debugging in the single user client server environment. We don't have any problem in a stand-alone environment. After adding users to the debugging group on the server, the problem still exists. The reason of this problem is not clear to the author.

6. RESOURCES

There is a huge amount of helpful online learning resources related to ASP.NET. Here are a couple of them, which are particularly helpful to the author.

http://www.aspalliance.com/aspxtreme/. Accessed April 17, 2003. This site provides many tutorials covering wide range ASP.NET concepts. They usually show you how to do a particular task step by step. Some of the examples have both C# and VB versions.

http://www.fawcette.com/resources/spcollections/aspnet/default.asp. Accessed April 17, 2003. This site provides many articles from intermediate level to highly technical level. These articles are mostly from online magazines and they discuss many interesting topics in ASP.NET.

http://www.codeproject.com/aspnet/. Accessed May 5, 2003. This site provides free source code and tutorials for developers. We can find complete examples for some typical tasks.

http://aspnet.4guysfromrolla.com/. Accessed May 5, 2003. This site provides wide range of tutorials for different levels of readers. This is my favorite site. I've been with it since Classic ASP. I found that whenever I meet a challenging problem, I always find a solution here.

http://www.aspnextgen.com/tutorialsindex.aspx. Accessed May 5, 2003. This site provides wide range of articles for different levels of readers. Articles are grouped according to topics, which is very helpful when we do research on a particular topic.

7. CONCLUSION

Moving from Classic ASP to ASP.NET has proven to be a challenging and exciting process. The author has learned a lot in this process. From the responses to our six-week survey, 100% students feel our WAD course challenging. However, most of them still prefer to learn ASP.NET rather than Classic ASP. We feel confident about the move. The issue is how to organize the course and help the students meet the challenge. ASP.NET is certainly an outstanding tool for both teaching and development. As a new development platform, we do need some time to absorb all the new features.

REFERENCES

Bloom, B. (2002) Leverage ASP.NET in Your Enterprise. .NET Magazine, July 2002 Issue. Accessed April 16, 2003. –HYPERLINK "http://www.fawcette.com/dotnetmag/2002_07/magazine/columns/architecting/" —http://www.fawcette.com/dotnetmag/2002_07/magazine/columns/architecting/

Goodyear, J. (2002) ASP.NET: What Matters, What Doesn't. Visual Studio Magazine January 2002 Issue. Accessed April 16, 2003. –HYPERLINK "http://www.fawcette.com/vsm/2002_01/online/online_eprods/asp_jgoodyear01_02/" —http://www.fawcette.com/vsm/2002_01/online/online_eprods/asp_jgoodyear01_02/

Kiely, D. (2003) Clean Out VSWebCach. Visual Studio Magazine January 2003 Issue. Accessed April 16, 2003. http://www.fawcette.com/vsm/2003_01/online/hottips/kiely/

Microsoft (2003) BUG: Named Pipes Do Not Work When Worker Process Runs Under ASPNET Account. Accessed May 5, 2003. http://support.microsoft.com/default.aspx?scid=kb;en-us;Q315159

Nair, C. (2002). Evaluation of subject, teaching and research. Proceedings of HERDSA 2002 (pp. 481-489), Perth, Australia, 7 - 10 July.

Olges, B. (2002) Simplify State Maintenance With ASP.NET. .Net Magazine October 2002 Issue. Accessed April 16, 2003. http://www.fawcette.com/dotnetmag/2002_10/online/bolges/