

Building Dynamic Web Applications in the Microsoft .NET Framework using a pure Model View Controller pattern

Csaba Bornemissza

The world of internet and intranet Web applications is a dynamic, rapidly changing area. Software products must follow the increasing needs as well as in quantity as in quality. New and new components and technologies appear to work with, and it is also common that we need to redesign our existing solutions.

With such a complicated background it is extremely important to build flexible applications, where the change of appearance and the continuous expansion are everyday tasks.

In this paper we will try to cover the vital aspects of developing reusable and easily redesignable Web applications. We will show the effectiveness of the well known Model View Controller design pattern. There are advantages and disadvantages when we follow a pre-made pattern; we show different situations of implementing this design pattern.

We will discuss some features of the Microsoft .NET Framework that are helpful to maintain software flexibility. Using scripting techniques like ASP.NET, or JSP will make the development process easier and quicker, however, in these scripts there is an unavoidable mixing of HTML and program code. Is it acceptable in a Model View Controller system? Can we call such a script-driven application a pure MVC application? Is the MVC approach preventing us to use all the advantages of the Web scripting techniques?

We try to answer the questions above. The XML technology became a popular way of inter-software communication. The XML Web Services give the Web developer community a standardized way of generating display-independent content. Using the XSL Transformations makes it possible to separate the information that is responsible for the presentation of the content (HTML, XHTML, or any format), and the code that generates the output data. We demonstrate how the XML Web Services and the XSLT help the developer to build flexible Web applications, where the software business Model, the Controller, and the View (end user displays, or inter-software interfaces) are perfectly separated, and independently maintained.

References

- [1] E. Gamma, R. Helm, R. Johnson, J. Vlissides. Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley, 1995.
- [2] C. Kerer and E. Kirda, Content and Logic Separation in Web Engineering. In 9th International World Wide Web Conference, 3rd Web Engineering Workshop, Amsterdam, Netherlands, May 2000.
- [3] Engin Kirda, Engineering of Web services with XML and XSL, September 2001 ACM SIGSOFT Software Engineering Notes , Proceedings of the 8th European software engineering conference held jointly with 9th ACM SIGSOFT international symposium on Foundations of software engineering, Volume 26 Issue 5
- [4] Microsoft .NET resources: <http://msdn.microsoft.com/asp.net>
<http://www.omg.org/cgi-bin/doc?formal/01-09-67> (2001)
- [5] World-Wide-Web Consortium, "Extensive Markup Language (XML)", At: <http://www.w3.org/XML>
- [6] World-Wide-Web Consortium, "The Extensible Stylesheet Language (XSL)", At: <http://www.w3.org/Style/XSL>
- [7] World-Wide-Web Consortium, "XSL Transformations (XSLT)", At: <http://www.w3.org/TR/xslt>
- [8] World-Wide-Web Consortium, XML Path Language (XPath) <http://www.w3.org/TR/xpath>