

Microsoft .NET Mobile Web

Mitko Gospodinov, Elmira Buchvarova, Andrei Zamanov

Central Laboratory of Mechatronics and Instrumentation, 1113 Sofia

E-mails: mitgo@abv.bg elmirabg@yahoo.com andrei@clmi.bas.bg

1. Introduction

There's a new trend in wireless telecommunications, Internet: Internet connectivity via handheld devices. More people are buying wireless handheld devices such as cellular phones, pagers, personal digital assistants (PDAs), and they demand access to online resources at any time from any location. Over the past few years a tremendous number of wireless devices different in their characteristics appeared on market. Developments of web application for these devices become challenging experience due to the following reasons [1]:

- Different markup languages are necessary, including HTML for PDAs, wireless markup language (WML) for wireless application protocol (WAP) cell phones, and compact HTML (cHTML) for Japanese i-mode phones.
- Different display capabilities – screen size (rows and columns) and color capability.

For example, a device with higher resolution can show more information and have better viewing quality. Some devices can display images and multiple colors, some can make phone calls, and others can receive notification messages.

- Devices have different network connectivity, ranging from 9.6-KB cellular connections to 11 MB Wireless LANs.

The Mobile Internet Toolkit is the solution that Microsoft has brought forward to help in mobile web development and overcome the difficulties with it.

2. The mobile internet toolkit overview

The Mobile Internet Toolkit makes fast and easy to build and maintain sophisticated mobile Web applications. It prevents developers from writing lots of custom code to

support many devices in their mobile Web applications. The toolkit extends the Microsoft Visual Studio.NET and the ASP.NET so that you can build a single mobile Web application that automatically adapts the display and interaction for a wide variety of mobile devices, including Web-enabled cell phones, pagers, and PDAs such as the Pocket PC [2]. The toolkit automatically generates the correct WML, HTML, and cHTML for the requesting device, providing the ability to support over 200 devices. Currently it supports WML 1.1, cHTML 1.0, and HTML 3.2.

3. The mobile Internet toolkit content

The Microsoft Mobile Internet Toolkit contains the following parts:

3.1. Mobile Internet Controls Runtime

It provides device-independent capabilities and extensibility for future device support. MICR consists of a set of ASP.NET server controls and device adapters that intelligently renders content for cell phones, PDAs and pagers [3]. The most important point is that controls do not render markup. Controls select an appropriate device adapter based upon the capabilities of a device. At a minimum, mobile controls need the following information about a device:

- Markup language (HTML, WML, cHTML);
- Browser;
- Number of display lines;
- Cookie support;
- Screen size.

The Mobile Internet Toolkit adds these mobile device capabilities to the server's "Machine.config" file. The device capability information in the "Machine.config" file allows device adapters to optimize a markup for specific devices [3].

3.2. Mobile Internet Designer

It integrates with Visual Studio.NET IDE, extends the graphical user interface of the Visual Studio.NET IDE to build easily mobile Web applications [4]. Using the designer, the developer can take advantage of the adaptive rendering, customization, and extensibility features in the Microsoft Mobile Internet Toolkit. It provides the standard Visual Studio IDE design tools: the Toolbox, forms creation, debugging capability, code windows (HTML View and Code View), interactive design (Design View), and more. Mobile Internet Designer facilitates the developer in the actions [4]:

- Create a mobile Web project;
- Add a mobile Web Forms page to the project;
- Drag a mobile Web Forms control onto the form;
- Double-click the control to write the logic;
- Rebuild the application;
- Run the application.

4. The mobile web application architecture

In this paragraph we will present the basic concept of how Microsoft Mobile Toolkit works. The following illustration (Fig. 1) shows outlines of Microsoft Mobile Toolkit mobile web application architecture.

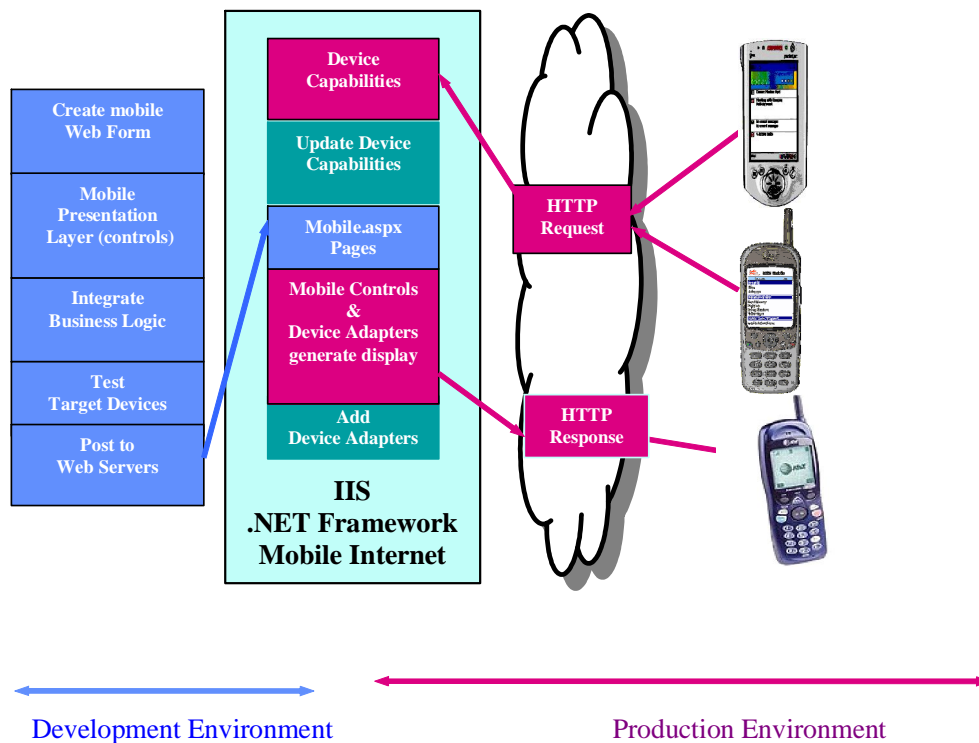


Fig. 1. The Mobile Web Application architecture

We will give a brief explanation about what happens at development environment and production environment. We will discuss what happens in development and production environment.

4.1. Development Environment

In each mobile Web application there are different logical layers: presentation, application, business logic, and data layer.

The presentation layer focuses on what the user will see, while the application layer focuses on how the application behaves and how the user can go from pages to pages. The business layer contains the components that manipulate and work on the data layer and also the components that understand the business domain in which the application lives. Finally, the data layer contains the data.

Let's preview what happens during the development process.

One of the first steps is creation of Mobile Web Form. This happens to be an “aspx” page, which is used to create ASP.NET development, with some special added features. Mobile Web Forms are server-side ASP.NET controls that are device independent typically controls. In this case controls rely upon a rendering engine, called the adapters, to do the rendering work that the controls would normally do [4].

The next step in development of mobile Web application is construction of a mobile presentation layer using the mobile controls, which integrates applied business logic. It could be Web services, data retrieval. The same business logic executed in mobile web application may be used by desktop Web applications allowing developers to re-use an existing code. After the mobile page is done, it is tested on the target devices or emulators and web application is deployed to the IIS Web server [1].

4.2. Production Environment

On the right side of production environment there is a Pocket PC and a cellular phone that want access to mobile Web application through HTTP request to IIS Web server. The HTTP request will be processed on the server in three main stages. First process is identifying the requesting device and its capabilities: browser, markup language and image capability. Microsoft Internet Toolkit extends .Net Framework “Machine.config” file with mobile device capability and pre-populates the device data. The HTTP request contains User Agent string, Header information and URL that is requested. The User Agent string is matched against entry in “Machine.config” file. The URL is used to locate the correspondent mobile web page-files with “aspx” extension. Parser parses the “aspx” page, after that it is compiled. The “aspx” page is compiled only one way, the first time it is accessed. The compiled page is stored in Assembly Cache. The servers then create a new instance of it and use it to process the request. The use of Assembly Cache improves performance. After the page is compiled, the page and controls over it are instantiated. The business logic contained on “aspx” page is executed. The device adapters associated with the requested device and controls on the page then generate appropriate mark-up language (Html, Wml). The mark-up language encapsulated in Http respond is returned to requested device (Pocket PC, WAP enabled cellular phones).

5. The real world

After discussion about Mobile Internet Toolkit web architecture we will outline some exciting mobile Web applications in which implementation is adopted .Net and Mobile Internet Toolkit. One of them is deployed by Scandinavian Airlines System, the consortium of the 3 Scandinavian airlines. SAS goal is to give customers the ability to check the status of flights and book tickets regardless of the user’s location, device, or time of day. By using MMIT to develop this web application, users with any internet-capable device have seamless access to this functionality. SAS have chosen to use MMIT because it enabled them to reach the widest possible range of mobile users with various cell phone and PDA devices while optimizing their development resources.

“Dollar Rent a Car” is another company that has taken the early initiative in deploying mobile Web application using MMIT and is benefiting from it greatly. Dollar Rent A Car started their .NET initiative by building an XML “web service” interface to their existing reservation system. Now, with MMIT, they are able to reuse this web service component in creating a large number of applications with the .NET platform as well as give third party developers the ability to integrate their applications with the “Dollar Rent A Car” reservations system. By using MMIT, “Dollar Rent A Car” became the first rental car agency to create a web site accessible from all wireless devices.

6. Conclusion

Although our paper can not cover all aspects of applying MMIT in the world of wireless Web, we believe that we have done our goal – to write a paper which provides a brief introduction to the development and deployment of Net mobile Web applications. We consider that the used MMIT technology is at the beginning of Microsoft initiative for development and deployment of web applications that will be accessed by any wireless device. It could be expected that the investigation of this field will gain rapid progress in the near future.

References

1. Milroy, S., Ken Cox. DotThatCom.com, Amit Kalani, Ken Cox, Wei Meng Lee “NET Mobile Web Developer’s Guide”, Syngress Media Inc., December 10, 2002.
2. Ferguson, D. Mobile .NET. 1st edition. APress, October 15, 2001.
3. Gibbs, M. ASP .NET Mobile Controls: Tutorial Guide: Adaptive Web Content for Mobile Devices with the MMIT. 1st edition. Wrox Press Inc., December 2001.
4. R R R R R R R R, R. .RRR RRRRRRRR RRRRRRRRRR. 1st edition. Sybex, February 2002.

Microsoft .NET мобильный Web

Митко Господинов, Елмира Бычварова, Андрей Заманов

Центральная лаборатория мехатроники и приборостроения, 1113 София

E-mails: mitgo@abv.bg , elmirabg@yahoo.com , andrei@clmi.bas.bg

(Резюме)

С широким распространением мобильного Internet и мобильных устройств, сегодняшние коммуникации и трансфер данных становятся все более мобильными. Почти все главные IT компании изучают рынок с точки зрения возможностей разработки мобильных технологий. В работе рассматривается Microsoft .NET Internet мобильная стратегия, представленная системой Microsoft Mobile Internet Toolkit API (MMIT). Это группа мобильного программного расширения, с помощью которой могут быть дополнены ASP.NET Web Forms. Используя эти модули, разработчики мобильных реализаций могут создавать мобильные системы, не беспокоясь об ограничениях целевого устройства.