

Software Platforms for Mobile Programming

Khoo Wei Ju

Malaysia University of Science and Technology, Malaysia

K. Daniel Wong

Malaysia University of Science and Technology, Malaysia

INTRODUCTION

Java 2 Micro Edition (J2ME), .NET Compact Framework (.NET CF), and Active Server Pages .NET (ASP.NET) Mobile Controls are commonly used alternatives in mobile programming. They provide an environment for applications to run on mobile devices. However, they are different in many ways, such as supported mobile devices, architecture, and development. Hence, it is important for mobile application developers to understand the differences between them in order to choose the one that meets their requirement. Therefore, in this article we will discuss the general architecture of J2ME, .NET CF and ASP.NET Mobile Controls and compare the three alternatives.

BACKGROUND AND INTRODUCTION

Since the mid-1990s, the growth of wireless communications has led to the mushrooming of mobile devices in the market. Initially, the mobile devices were mainly cell phones with limited programmability. However, many analysts and company executives were worried that mobile phone sales would eventually slow down, prompting research and development into software suitable for cell phones (Grice & Charny, 2001). Hence, now, there is a rise of programmable mobile devices. Furthermore, programmable mobile devices these days include not just cell phones but smartphones, PDAs, and pocket PCs. There are three well-known alternatives in mobile programming for general-purpose applications: J2ME, .NET CF, and ASP.NET Mobile Controls.

J2ME is a version of Java that provides an application environment running on consumer devices and embedded devices. It targets machines with as little as 128KB of RAM (Tauber, 2001). J2ME consists of Java virtual machines (JVMs) and a set of standard Java application program interfaces (APIs) defined through the Java community process (JCP). J2ME can be used with different configurations and profiles, which provide specific information to a group of related devices. Configurations support the Java core APIs. Profiles are built on top of configurations to support device-specific features like networking and user interfaces. The J2ME is available in two main configurations: connected

limited device configuration (CLDC) and connected device configuration (CDC). Figure 1 shows the hierarchical structure of J2ME.

.NET CF is a lightweight version of Microsoft's .NET framework. It provides an environment for executing client-side code and eXtensible Markup Language (XML) Web services to smart devices. It is compatible with C# and Visual Basic.NET (VB.NET), and it supports (.NET Compact Framework Team, 2005):

- Windows mobile (2000, 2002, 2003)-based pocket PC,
- Windows mobile-based smartphones, and
- embedded systems running Windows CE .NET 4.1 and later.

.NET CF consists of two main components: the development environment and the runtime environment. The development environment, known as smart device extensions (SDEs), is a Visual Studio .NET (VS.NET) 2003 project type that allows .NET CF applications to be developed rapidly by simply dragging appropriate controls into the application. The runtime environment is the common language runtime (CLR). The size of the CLR and relevant class libraries is smaller than 2MB, which is suitable for mobile devices. The architecture of .NET CF is shown in Figure 2.

Active server pages (ASP) is Microsoft's server-side scripting technology. An active server page has an .asp extension, and it mixes HyperText Markup Language (HTML) and scripting code that can be written in VBScript or JavaScript. ASP is distributed with Microsoft's Internet information services (IIS) Web server, so most hosts using IIS will also offer ASP for dynamic Web programming. ASP.NET is the version of ASP that works with Microsoft's .NET Framework.

ASP.NET Mobile Controls was previously known as Microsoft mobile Internet toolkit (MMIT). It was renamed as ASP.NET Mobile Controls to reinforce the concept that it is a collection of ASP.NET controls designed for mobile applications. It extends the ASP.NET server-side technology to allow developers to develop applications for a variety of mobile devices. Executing on the IIS Web server, ASP.NET Mobile Controls allows Web applications to be accessed by

Figure 1. Hierarchical structure of J2ME

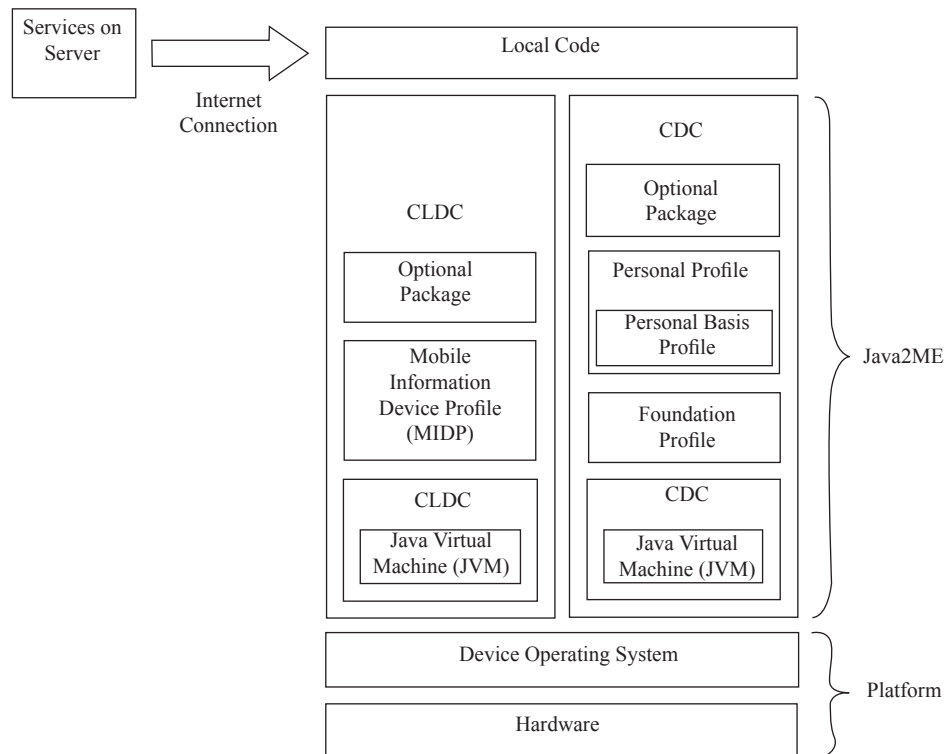
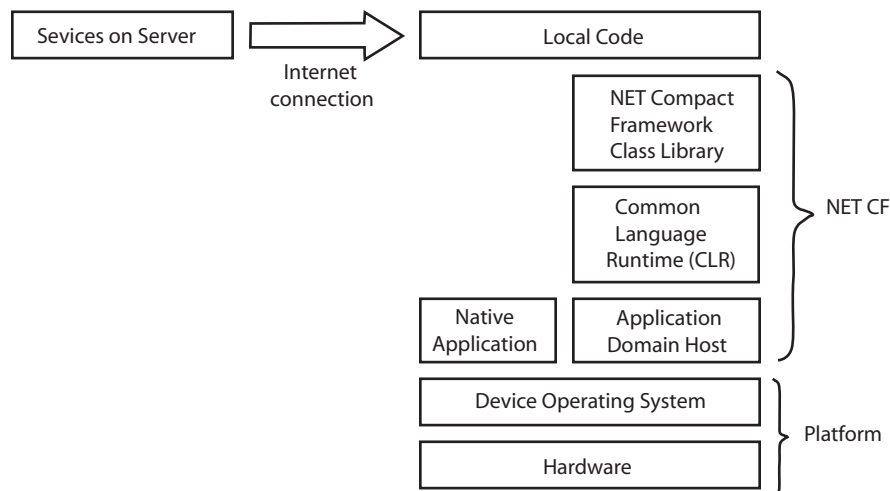


Figure 2. .NET compact framework



almost any Internet-enabled mobile device. During runtime, it will automatically detect the device running the application. The application is then transformed into a form suitable for that device. This frees the developer to concentrate on the application logic and leaves the user interface rendering to the runtime (Lee, 2002a). Furthermore, it allows developers to visually drag and drop controls on forms aimed at mobile

devices using VS.NET. The rest of the work, such as writing the proper markup language (e.g., Wireless Markup Language (WML), wireless application protocol (WAP)), is handled by the toolkit. The application development environment for ASP.NET Mobile Controls should be familiar to most ASP.NET programmers. Figure 3 shows the architecture of ASP.NET Mobile Controls.

7 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage: www.igi-global.com/chapter/software-platforms-mobile-programming/17195

Related Content

Mobile Automotive Cooperative Services (MACS): Systematic Development of Personalizable Interactive Mobile Automotive Services

Holger Hoffman, Jan Marco Leimeister and Helmut Krcmar (2009). *Mobile Computing: Concepts, Methodologies, Tools, and Applications* (pp. 1499-1515).

www.irma-international.org/chapter/mobile-automotive-cooperative-services-macs/26603

Effect of Personal Innovativeness, Attachment Motivation and Social Norms on the Acceptance of Camera Mobile Phones: An Empirical Study in an Arab Country

Kamel Rouibah and Hasan A. Abbas (2010). *International Journal of Handheld Computing Research* (pp. 41-62).

www.irma-international.org/article/effect-personal-innovativeness-attachment-motivation/48503

Residual Reconstruction Algorithm Based on Half-Pixel Multi-Hypothesis Prediction for Distributed Compressive Video Sensing

Ying Tong, Rui Chen, Jie Yang and Minghu Wu (2018). *International Journal of Mobile Computing and Multimedia Communications* (pp. 16-33).

www.irma-international.org/article/residual-reconstruction-algorithm-based-on-half-pixel-multi-hypothesis-prediction-for-distributed-compressive-video-sensing/214041

Mobile-Assisted Language Learning: Research-Based Best Practices for Maximizing Learner Success

Katharine B. Nielson (2016). *Handbook of Research on Mobile Learning in Contemporary Classrooms* (pp. 69-93).

www.irma-international.org/chapter/mobile-assisted-language-learning/157975

Biocompatible Implanted Dielectric Sensors for Breast Cancer Detection

Noah P. Svoboda and Abas Sabouni (2014). *International Journal of Handheld Computing Research* (pp. 1-19).

www.irma-international.org/article/biocompatible-implanted-dielectric-sensors-for-breast-cancer-detection/137117