

Chapter 3.7

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):

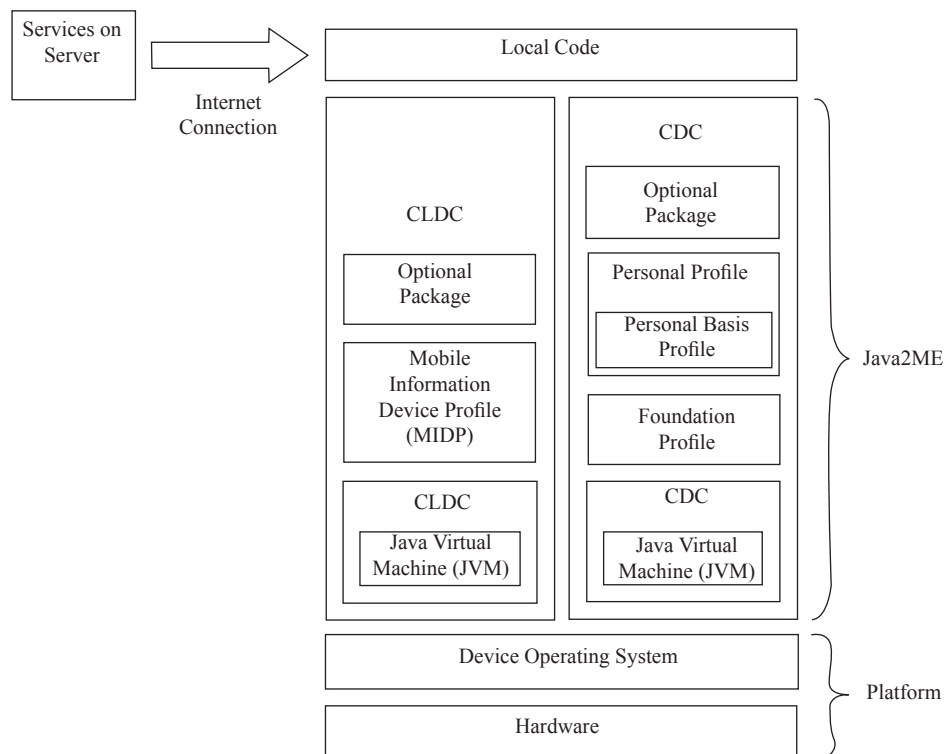
- Windows mobile (2000, 2002, 2003)-based pocket PC,
- Windows mobile-based smartphones, and

- 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 (ASPs) is Microsoft's server-side scripting technology. An active server page has an .asp extension, and it mixes HyperText

Figure 1. Hierarchical structure of J2ME



10 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/29443

Related Content

On the Accelerated Convergence of Genetic Algorithm Using GPU Parallel Operations

Cheng-Chieh Li, Jung-Chun Liu, Chu-Hsing Lin and Winston Lo (2015). *International Journal of Software Innovation* (pp. 1-17).

www.irma-international.org/article/on-the-accelerated-convergence-of-genetic-algorithm-using-gpu-parallel-operations/133111

Software Review Tools and Technologies

Yuk Kuen Wong (2006). *Modern Software Review: Techniques and Technologies* (pp. 37-52).

www.irma-international.org/chapter/software-review-tools-technologies/26900

Agile Development Processes and Knowledge Documentation

Eran Rubin and Hillel Rubin (2014). *Handbook of Research on Emerging Advancements and Technologies in Software Engineering* (pp. 309-328).

www.irma-international.org/chapter/agile-development-processes-and-knowledge-documentation/108623

Cloud Computing Decisions in Real Enterprises

Manuel Pérez-Cota, Ramiro Gonçalves and Fernando Joaquim Lopes Moreira (2014). *Agile Estimation Techniques and Innovative Approaches to Software Process Improvement* (pp. 313-330).

www.irma-international.org/chapter/cloud-computing-decisions-in-real-enterprises/100285

Observer Patterns for Timed Properties

Djamila Baroudi and Safia Nait-Bahloul (2021). *International Journal of Software Innovation* (pp. 1-17).

www.irma-international.org/article/observer-patterns-for-timed-properties/277212