

Chapter 14

A CASE Tool for Java Mobile Computing Applications

Ioannis T. Christou

Athens Information Technology, Greece

Sofoklis Efremidis

Athens Information Technology, Greece

Aikaterini Roukounaki

Athens Information Technology, Greece

ABSTRACT

The market for applications developed for mobile devices is growing as the hardware capabilities increase while costs drop. At the same time, the inability to write code once becomes a noticeable problem resulting in the time consuming task of porting an application to a variety of mobile devices, which is true for languages like Java that have been designed to be portable across a range of hardware platforms. As a result, porting a Java application written for one device to another is often a tedious and time-consuming task for developers. This paper presents an intelligent CASE tool that assists the porting of Java mobile applications by automatically generating Java code fragments for the target SDK. SeqFinder automatically generates all minimal method invocation sequences that lead to an object of a specific type, thus relieving the programmer of manually searching the manufacturer-provided SDK Java archives. However, this tool is not applicable only to the Java ME platform and has been used as a fast type-browser for J2SE/J2EE applications.

INTRODUCTION

Mobile Computing has witnessed significant growth in the past two decades, since its initial conceptualization. While the focus of attention has often been the mobile laptop user with roaming

WIFI connections (Satyanarayanan, 1996; Bellavista et al., 2001; Pittoura & Samaras, 2001), lately there has been significant focus on developing applications for constrained mobile devices such as smart-phones and Personal Digital Assistants (PDAs) (Sohn et al., 2006; Tomek & Shakshuki, 2009; Sugumaran et al., 2009; Rawassizadeh, 2009).

DOI: 10.4018/978-1-4666-0119-2.ch014

Application development has always been a challenging task that becomes even more difficult for applications targeting mobile devices where porting issues often prevail. The inability to write once, run everywhere can be attributed to two main reasons:

1. Hardware diversity, i.e., differences in hardware characteristics among mobile devices, like screen size, color depth, memory size, support for peripheral devices like cameras, Bluetooth connectivity, etc.
2. Software diversity, which refers to
 - a. Differences in the underlying operating system (Symbian, PalmOS, etc.)
 - b. Differences in the supported standards, i.e., supported versions of the Connected (Limited) Device Configuration (“Connected Limited Device Configuration (CLDC)”, 2009) and Mobile Information Device Profile (“Mobile Information Device Profile (MIDP)”, 2009; Ortiz & Giguere, 2001) of the Java 2 Mobile Edition (“Java ME – The Most Ubiquitous Platform for Mobile Devices”, 2009)
 - c. Differences in the implementation, i.e., vendor specific implementations of MIDP or CLDC APIs, or even vendor/device/series specific libraries and APIs extending or complementing those specified in the standard CLDC and MIDP, for example codecs, graphics support APIs, etc.

It turns out that this diversity is rather difficult to mask off from application developers. Contrary to the desktop and server world, where Java Virtual Machine are almost universally based on SUN’s Java SDK (j2sdk) regardless of Operating System and computer architecture of the machine running the JVM, the Java Kilobyte Virtual Machines (KVM) installed in the mobile phones of today are the manufacturer’s implementation of the KVM

specification. This practically means that each mobile device runs a different implementation of the same specification than any other mobile device of another manufacturer. Implementation bugs, combined with the fact that the specification does not always specify completely what a functionality or Application Programming Interface (API) is supposed to do, have lead to an application development and maintenance nightmare for third-part developers of mobile applications. It is an unfortunate fact therefore, that the promise of Java (Gosling, Joy, Steele, & Bracha, 2000) as a “write once, run everywhere” language proved to be overly optimistic in the mobile phone market-place. This is by no means a criticism on the language itself, as Java is probably the best-suited language for writing applications for mobile phones and mobile information devices in general, but rather a realization that the extreme variation in those devices’ characteristics necessitated different versions of the language specifically tailored to these devices. Undoubtedly, the area with the most variations in the offered APIs is the Graphical User Interface, but as different devices support different features and offer the corresponding APIs for developers to take advantage of, there is a wide range of capabilities of different devices and, correspondingly, of the APIs supporting them.

Even standardized APIs such as the connectivity API that allows access to the Internet and servers and services available over HTTP or socket protocols are often poorly implemented by the device manufacturers and cause serious porting problems to application developers. To further complicate things, many (early) phone models that claimed to support the MIDP specification had serious bugs in the built-in Kilobyte Virtual Machine (KVM) especially regarding synchronization issues, thread management, and memory management. These defects combined, make writing complex applications for such devices at times a daunting task.

12 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/case-tool-java-mobile-computing/62974

Related Content

The Design of Mobile Television in Europe

Pieter Ballonand Olivier Braet (2009). *Mobile Computing: Concepts, Methodologies, Tools, and Applications* (pp. 1143-1167).

www.irma-international.org/chapter/design-mobile-television-europe/26577

Improving the Security of Storage Systems: Bahrain Case Study

Wasan Shaker Awadand Hanin Mohammed Abdullah (2014). *International Journal of Mobile Computing and Multimedia Communications* (pp. 75-105).

www.irma-international.org/article/improving-the-security-of-storage-systems/130482

Security Management for Mobile Ad Hoc Network of Networks (MANoN)

Ali H. Al-Bayatti, Hussein Zedan, Antoniuo Cauand François Siewe (2010). *International Journal of Mobile Computing and Multimedia Communications* (pp. 1-19).

www.irma-international.org/article/security-management-mobile-hoc-network/40978

Smartwatch-Based Data Analytics and Feature Selection for Heart Failure Assessment

Xu-Jun Jian, Chao-Hung Wang, Tieh-Cheng Fu, Shiyang Lyu, David Taniarand Tun-Wen Pai (2025). *International Journal of Mobile Computing and Multimedia Communications* (pp. 1-13).

www.irma-international.org/article/smartwatch-based-data-analytics-and-feature-selection-for-heart-failure-assessment/371205

Mobile Apps, Universal Design, and Accessibility in Schools: Creating an Inclusive Classroom Experience

Tseday Alehegnand Richard Keller (2019). *Advancing Mobile Learning in Contemporary Educational Spaces* (pp. 317-343).

www.irma-international.org/chapter/mobile-apps-universal-design-and-accessibility-in-schools/234057