SOFTWARE DISTRIBUTION FOR WIRELESS DEVICES
A reconfigurable approach

Gráinne Foley and Fergus O'Reilly
Cork Institute of Technology, Bishopstown, Cork, Ireland

Abstract: The dramatic rate of evolution and technology divergence from 2G to 3G and onwards broadens requirements for interoperability across software systems and wireless devices. These should interoperate seamlessly, transparent to users, who need only concern themselves with the final result. The challenge of providing transparent and reconfigurable content to wireless devices is addressed in this paper.

Key words: Wireless networks, Software distribution, Reprogrammable wireless devices

1. INTRODUCTION

Currently most European countries offer GSM and GPRS networks. Roaming on the GSM network is widespread and popular through a series of cooperating standards and agreements. The same system does not operate worldwide. In order to bridge the gaps and advance wireless telecommunications, 3rd generation (3G) networks were envisaged [1]. These networks require changes in network structure, user devices – hardware and software implementations. With the continued rapid development and rollout of wireless IP infrastructures and devices, the rate of obsolescence of user equipment is increasing dramatically. Currently 25% of handsets are replaced annually [2]. This trend will accelerate as new network architectures and technologies are introduced. Maturing these technologies will require many iterations and releases of software and hardware.

A system for software and device independence is explored here using Java. Java provides platform independence and is ideally suited to software delivery to divergent wireless devices over differing networks. It is predicted that Java will be present in 74% of wireless phones shipped in 2007 [3]. Java is implemented in the Services Archive (see below), making the solution presented Java end-to-end.

2.   SERVICES ARCHIVE ARCHITECTURE

The Heterogeneous Services Archive illustrated in Figure 1 manages the distribution of software and firmware/hardware updates to wireless devices.
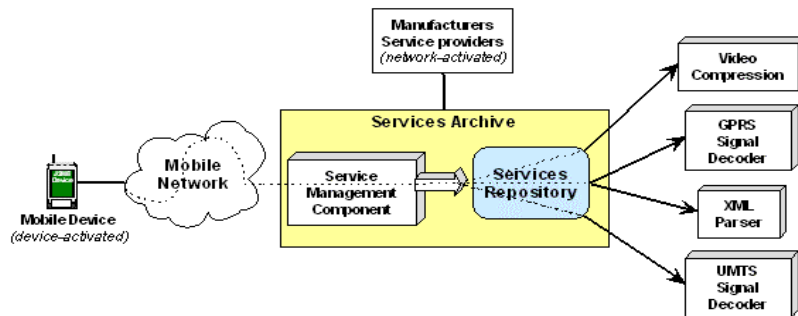

Figure 1. Heterogeneous Service Archive Architecture

User services may be device-activated, on a user service requirement basis, or network-activated, facilitating operators, manufacturers and service providers [4]. Thus, the mobile device must be able to support dynamic service downloading facilities, in addition to service execution. Two Java platforms are employed, the Java 2 Enterprise Edition (J2EE) in the Services Archive and the Java 2 Micro Edition (J2ME) on the wireless device [5].

3.   WIRELESS DEVICE ARCHITECTURE

J2ME provides a hardware independent platform designed for limited memory footprint devices. It does this using the Connected Limited Device Configuration (CLDC) and the Mobile Information Device Profile (MIDP) [5], which respectively provide virtual machine features needed to support wireless mobile devices; and networking, user interface, persistent storage and application APIs. Code for the MIDP and CLDP resides in a vendor-customed Kilobyte Virtual Machine (KVM). It is suitable for devices with 16/32-bit RISC/CISC microprocessors/controllers, with as little as 512 KB of total memory available. The J2ME wireless device has a four-component architecture:

a) Management Component: Managing software and hardware components.
b) Services Component: Controls the execution of MIDP applications – MIDlets. A MIDlet, or a suite, is packaged in a Java Archive file (JAR) containing code and other resources. There may be many MIDlet suites on a wireless device.
c) Persistent Memory Component: For application and versioning data.
d) Hardware Component: Implemented using an Adaptive Computing Machine (ACM) [6]. The ACM provides advantages over other hardware technologies (DSP, ASIC and FPGA) enabling dynamic fast reconfiguration of evolving algorithms and standards without concerns of untimely obsolescence [7].

4. WIRELESS DEVICE UPDATES

Software updates are largely catered for within the J2ME specification. Updates to the underlying hardware are delivered from the Services Archive to the wireless device, Figure 2. Suitability is firstly verified by downloading a Java descriptor file.
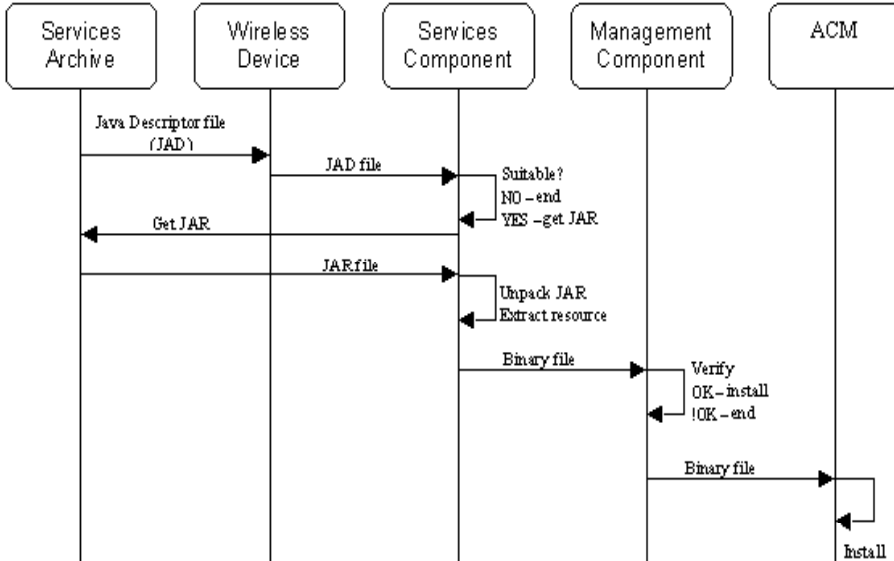

Figure 2. Hardware updates sequence diagram

The services component extracts an ACM resource (binary) from a JAR file. The management component then checks and installs the file, an MPEG-4 update for example. This mechanism requires operating system support and interaction but makes extendible, reconfigurable component based mobile devices a reality.

5. MEMORY AND PERFORMANCE EVALUATION

The MIDP specifies 32KB of run time memory for all Java applications. The statistics shown in Table 1 use the J2ME reference implementation.

Table 1. Execution statistics for MIDlet suites (bytes)

MIDlet Suite	Details	JAR file size	Executed byte codes	Dynamic objects	Heap Used
MyFileClient	Connect server, get file	3,266	120,182	43,664	35,020
Animation	Series- PNG images	12,690	336,718	64,488	36,080
WTK Demo	Show system properties	144,445	276,502	93,368	32,608

The JAR file is highly compressed for efficient delivery. It swells significantly at run time indicating that even very small suites may use considerable amount of

memory. The amount of heap used for one MIDlet execution already exceeds 32KB in two cases. While manufacturers can optimize memory usage, they are already targeting total heap sizes in excess of 32KB. The Motorola iDEN phones specify a maximum JAR size of 50KB for one MIDlet suite, while the Nokia 3410 specifies 30KB.

Table 2. Typical download times for MIDlet suites with GPRS and UMTS

MIDlet Name	Function	Size (KB)	Transfer Time	
			GPRS*	UMTS^
Nokia Tester	For testing Nokia phone features	5KB	0.19 sec	0.05 sec
Crossword Solver	Crossword puzzle solver	6KB	0.23 sec	0.06 sec
Live Weather	Graphical current weather	25KB	0.94 sec	0.25 sec
SoccerLeague	Online multiplayer soccer	88KB	3.3 sec	0.88 sec
Uemail	Mail for J2ME and WAP clients	164KB	6.2 sec	1.64 sec

* GPRS two channels – 26.8kbps   ^ UMTS at likely 100kpbs throughput

From Table 2 we can see limits in Over-The-Air capacity may not significantly limit MIDlet suite size and functionality. If roaming from 3G to GPRS, performance rates are still reasonable. This is important, as JAR sizes will increase to deliver new services to devices.

6.   CONCLUSIONS

In this paper we examined an overall architecture to provide transparent and reconfigurable content to wireless devices. The Services Archive manages the distribution of services activated from the network or from the device itself. The wireless device employs Java as a delivery mechanism for updates to software and firmware/hardware. Wireless devices, reconfigurable OTA, become more flexible and robust in a rapidly changing personal communications environment. This reduces the cost of technology evolution, an important consideration given the significant concerns over 3G-rollout expenditure [8].

REFERENCES

1. J. De Vriendt, P. Lainé, C. Lerouge, X. Xiaofang, "Mobile Network Evolution: A Revolution on the Move," *IEEE Communications*, Apr. 2002, pp 104-111
2. Nokia CEO, Jorma Ollila at Year End Strategy Update, December 3, 2002 Dallas Texas
3. Zelos Group Report "Wireless Java (Sept. 2002)" http://www.zelosgroup.com
4. Farnham, T., Clemo, et al.., 'Reconfiguration of Future Mobile Terminals using Software Download', IST Mobile Communications Summit 2000. pp.159- 168
5. J2EE and J2ME at http://java.sun.com/. Specifications at http://jcp.org/
6. Tuttlebee, W. (Ed.), *Software Defined Radio Enabling Technologies*, J Wiley & Sons Ltd, Chichester, England, 2002, pp. 272 - 288
7. Bing, B., Jayant, N. "A Cellphone for all Standards," *IEEE Spectrum*, May 2002, pp 34-39
8. "Ericsson sends shock waves across sector" Financial Times, 22/4/2002