

모바일 플랫폼의 기술현황 및 발전방향

삼성전자 홍준성

1. 서 론

현재 휴대폰 시장은 다른 어떤 시장보다 급격하게 성장 가도를 달리고 있다. PDA 형태에 통화 기능을 부가한 제품이 예상 외로 고전하고 있는데 반해, MMS (Multimedia Messaging Service)나 카메라 촬영 등의 멀티미디어 기능이 강화된 휴대폰이나, 상대적으로 큰 LCD와 펜/키보드 입력 등의 정보 처리 기능이 강화된 휴대폰이 성장세를 이어가고 있다. 전자의 경우 1세대인 음성 중심의 휴대폰에 WAP browser 등을 탑재한 2세대 데이터 중심의 휴대폰을 거쳐 비디오 촬영, MMS, 화상 통화 기능 등을 탑재한 2.5세대~3세대 멀티미디어 중심의 휴대폰으로 발전해 나가고 있고, 후자의 경우 PDA 수준의 정보 처리 기능을 기반으로 하여 멀티미디어 기능을 계속 추가하면서 발전해 나가고 있다. 이러한 차세대 휴대폰의 중심에는 모바일 플랫폼이 자리잡고 있다.

모바일 플랫폼이란 모바일 기기에 탑재될 수 있는 기능/크기/성능을 가진 잘 정의된 API 및 개발 환경을 제공하는 소프트웨어 시스템을 말한다. 실제로 모바일 플랫폼이라는 용어는 하드웨어에서 소프트웨어까지 매우 광범위하게 사용되고 있어서 엄밀히 말하면 모바일 소프트웨어 플랫폼이라고 칭하는 것이 더 정확한 표현일 수 있으나, 이 글에서는 모바일 플랫폼을 소프트웨어 솔루션에 한정하고, 나아가 휴대폰 기능을 기본 기능으로 가지는 모바일 기기에 적합한 플랫폼에 한정하여 설명하도록 한다.

모바일 플랫폼을 다시 나누어 보면, 애플리케이션의 OTA(Over-The-Air) 다운로드가 주요 목표인 플랫폼과 단말기 개발이 주요 목표인 플랫폼으로 크게 구분할 수 있다. 전자의 경우 보통 무선 인터넷 플랫폼이라고도 불리며, 이 부류에 속하는 흔히 알려진 플랫폼은 Sun Java, QUALCOMM BREW, WIPI 등이 있다. 이 종류의 플랫폼은 주로 사업자의 수익에 큰 영향을 주는 플

랫폼이므로 사업자가 선정하여 단말기 업체에게 제공하는 형태가 일반적이다. 후자의 경우 흔히 알려진 플랫폼은 Microsoft Windows Mobile for Smartphone, Symbian OS, PalmOS, Ericsson Mobile Platform 등이 있으며 이 종류의 platform은 주로 휴대폰 개발 전략에 큰 영향을 주는 플랫폼이므로 단말기 업체가 도입 혹은 개발하는 형태를 가지게 된다.

모바일 애플리케이션의 경우 2001년 정도까지만 해도 browser가 유일한 애플리케이션이었고 Openwave, Nokia, AU, 삼성전자 등에서 browser를 개발하여 데이터 중심적인 휴대폰에 적용하여 왔다. 하지만 MMS, Java, Instant Messaging, SyncML, DRM 등 다양한 애플리케이션 및 미들웨어가 필요하게 됨에 따라 이러한 기능들을 하나로 애플리케이션 스위트화 하여 단말기 업체에 판매하려는 전략이 Openwave, Teleca, Magic4 등 여러 업체에 의해 가속화 되고 있다.

모바일 플랫폼의 진화 방향을 살펴보면 file system, WAP, window system 등 플랫폼의 기본 기능 외에 browsing, messaging, phonebook, SyncML, DRM 등과 같은 애플리케이션 레벨의 공유 서비스를 계속 플랫폼에 추가하여 영역을 확대함으로써, 플랫폼에 UI 컴포넌트들을 추가하면 하나의 휴대폰 제품이 될 수 있게 한다. 이에 반해 모바일 애플리케이션의 진화 방향은 다양한 애플리케이션에 공통적으로 사용되는 서비스들을 미들웨어화 하여 애플리케이션 스위트 형태로 제공함으로써, 미들웨어를 탑재하면 애플리케이션들을 적은 노력으로 휴대폰에 탑재할 수 있게 한다. 따라서 최근에는 모바일 플랫폼과 모바일 애플리케이션 스위트의 구분이 모호해지고 있는 상황이다.

이 글에서는 간략하나마 휴대폰에 탑재되는 모바일 플랫폼과 연관된 소프트웨어 기술을 전반적으로 다룬다. 표 1은 여러 다양한 모바일 플랫폼과 애플리케이션 스위트에 대해 정리한 것이며 이 글에서는 이 중 일부 플랫폼의 기술적 관점을 살펴보도록 한다.

표 1 모바일 플랫폼의 분류

분류	이름	기술 주도/업체	주 개발언어	비고
무선 인터넷 기반 모바일 플랫폼	Java	Sun Microsystems, JCP	Java	Virtual machine
	BREW	QUALCOMM	C/C++	Binary code
	MAP	(주)모빌탑	C/C++	Binary code
	WIPI	무선 인터넷 표준화포럼, TTA	C/C++, Java	Binary code
	GVM	신지소프트	Mini C	Virtual Machine
단말기 제조 기반 모바일 플랫폼	Windows Mobile for Smartphone	Microsoft	C/C++	
	Symbian OS	Symbian	C++, Java, OPL	
	PalmOS	PalmSource	C	
	Embedded Linux	CE Linux Forum, MontaVista 등	C/C++	
	Mocha	삼성전자	C	
	EMP	Ericsson Mobile Platforms	C	
애플리케이션 스위트	Openwave	Openwave	C	
	TSS, Series 60	Nokia	C	
	Obigo	Teleca	C	
	Magic4	Magic4	C	

2. 무선 인터넷 기반 모바일 플랫폼

무선 인터넷 기반 모바일 플랫폼은 음성 혹은 데이터 중심적인 휴대폰에 사용되는 대부분의 소프트웨어 및 주요 애플리케이션이 탑재된 단말기에 하나의 애플리케이션으로 추가적으로 이식되어 플랫폼 애플리케이션을 수행할 수 있도록 수행 환경을 제공하는 플랫폼을 말한다. 즉, 플랫폼의 크기는 코드 크기가 보통 최대 1MB 정도 이내에 들 정도로 작으며 최근에 browser, instant messenger 등을 플랫폼 위에서 수행하는 방향으로 애플리케이션의 범위를 확장해 가고는 있으나 아직까지는 주로 게임, 모바일 뱅킹 등의 특화된 애플리케이션을 위해 주로 사용된다.

2.1 Java

현재 세계적으로 가장 널리 사용되는 모바일 플랫폼에는 Java 기술을 모바일에 적용한 J2ME (Java 2 Micro Edition)가 있다. Nokia는 2002년 Java를 탑재한 휴대폰을 5천만대 가량 출시할 것이라고 예고해 실제로는 천만대 가량의 출시에 그쳤으나, Java를 중심으로 한 전략은 계속되고 있고 유럽과 일본을 중심으로 많은 사업자들은 Java를 모바일 플랫폼으로 선정하고 있다.

J2ME는 단말기의 형태에 따라 CDC (Connected Device Configuration) 기반과 CLDC (Connected Limited Device Configuration) 기반으로 나누어진다. 기존에 시판되는 휴대폰에서는 KVM 위에 CLDC와 MIDP1.0 (Mobile Information Device Profile)이 주로 사용되고 있으나 표준 profile의 기능 부족을 이유

로 각 사업자마다 고유의 OEM 확장을 하는 경우가 많다. (예 : Vodafone, SprintPCS, J-Phone) 하지만 급년에 시판될 휴대폰에는 점차적으로 MIDP 2.0 (JSR-118)을 기본으로 하고 추가적으로 Wireless Messaging API (JSR-120), Mobile Media API (JSR135), Mobile 3D Graphics (JSR-184) 등의 API를 탑재하게 될 것으로 보인다.

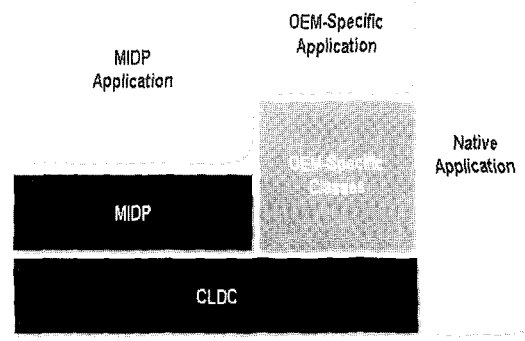


그림 1 J2ME (CLDC/MIDP) 구조

그림 1은 J2ME 중 CLDC/MIDP의 구조를 보여준다. CLDC는 Java의 기본 class들을 제공하며 이 위에 MIDP와 같은 profile이 추가되어 MIDP 애플리케이션이 수행되게 된다. OEM 확장을 하는 경우는 OEM-specific class들이 그림과 같이 추가 가능하며 OEM specific application을 수행할 수 있다. 이와 같은 Java 실행 환경은 하부의 native system software 위에 이식되며 native application과 같이 수행 가능하다.

Java와 관련된 기술은 JCP (Java Community Process)에 의해 표준이 정의된다. 실제 구현은 여러 회사에 의해 독자적으로 이루어지고 있으며 단말기 업체에서는 Sun에서 제공하는 reference implementation을 최적화하거나, 이와는 달리 cleanroom 방식에 의해 개발한 QUALCOMM QVM, HP MicroChai VM, XCE XVM 등을 사용한다. 국내의 경우를 예로 들면 무선 인터넷 서비스 초기인 2001년 8월에 삼성전자 SCH-X350에 탑재되어 SKT에 의해 서비스된 XVM은 J2ME의 KVM을 cleanroom 방식으로 구현한 것으로 CLDC를 구현한 M-Configuration (이후 C-Configuration으로 변경), MID Profile, 그리고 SKT API로 구성되어 있다.

Java의 가장 큰 단점은 bytecode를 실행 중에 해석하는데 따르는 성능 저하라고 할 수 있다. 이에 대한 보완책으로 각광 받는 소프트웨어 적으로는 JIT (Just In Time) compilation, AOT (Ahead of Time) compilation 등의 기법을 사용하거나 interpreter를 어셈블리 언어로 최적화하는 등의 방법이 있고, 하드웨어 적으로는 전용 가속기를 기존 chipset에 추가적으로 사용하거나 가속기를 애플리케이션 프로세서나 모뎀 칩에 같이 탑재하여 사용하는 방법도 있다.

2.2 BREW

BREW (Binary Run-time Environment for Wireless)는 QUALCOMM chipset을 사용하는 단말기 개발자가 공통의 소프트웨어 모듈 및 애플리케이션을 사용할 수 있도록 하기 위해 QUALCOMM이 개발한 binary code 방식을 지원하는 모바일 플랫폼이다. 2001년 11월 KTF를 통해 최초 상품화에 성공하였고 이후 Verizon Wireless, KDDI, BellSouth 등 10여 개의 사업자가 BREW 기술을 채택하였다. 현재까지 BREW 3.0이 나와 있으며 BREW1.0에 대비 Bluetooth, Browser component, Image CODEC, CMX2.2 interface, Voice, Graphics, Secondary display access, MMC/SD directory access, Generic serial interface 등의 기능이 추가되었다.

BREW 애플리케이션은 C/C++을 사용하여 개발을 하게 되며 실행 코드가 binary code 형태를 가지므로 Java에 비해 10배까지도 성능이 빠르고, 플랫폼 API 추가 및 구현을 QUALCOMM이 독자적으로 진행하므로 JCP에 비해 version up의 속도가 빠르다는 등의 장점이 있다. 하지만 실행 코드의 크기가 Java에 비해 두 배 가량 크고, BREW engine의 탑재가 QUALCOMM chipset으로 한정되며, 보안 모델이 취약하다는 등의 단점을 가지고 있다. 예를 들어 현재 휴대폰에 사용되는

주요 chipset에는 주로 MMU (Memory Management Unit)가 없는 경우가 많으며 이 경우 애플리케이션의 잘못된 포인터 연산 등으로 인해 휴대폰의 안정성에 문제가 생길 수 있는 가능성이 있어 certification등의 노력을 한다고 해도 구조적으로는 Java에 비해 잠재적 불안정성의 원인을 가지고 있다고 할 수 있다.

2.3 MAP

MAP은 (주)모빌탑이 국내 기술로 개발한 Mobile Application S/W Plug-in이다.¹⁾ Java의 bytecode는 이식성/호환성이 뛰어난 반면 성능 측면에서 한계가 있기 때문에 대신 C언어를 사용하여 binary code를 다운로드 하도록 함으로써 수행 성능 측면에서 장점을 가지는 플랫폼이다. 현재는 BREW에 의해 이 방식이 널리 알려졌으나 MAP은 2001년 3월에 KTF, 한국통신 엠닷컴을 통해 상품화가 되었다는 점에서 이 방식의 시초라고 할 수 있다. 이후 KTF용 휴대폰에는 기본적으로 탑재되어 오다가 작년에는 애플리케이션이 구동시키는 wake up 기능 등이 추가된 MAP 2.0을 통해 해외 시장 공략에 나서고 있다.

2.4 WIPI

2001년 중반 이후 개발이 시작된 국내 무선 인터넷 플랫폼 표준인 WIPI (Wireless Internet Platform for Interoperability)는 2002년 말부터 국내의 각 사업자에 채택되어 올해 국내에서 시판되는 휴대폰의 주요 모바일 플랫폼이 될 것으로 예상된다. 특히 WIPI는 단말기 제조 업체나 사업자가 단독으로 개발한 것이 아니라 한국 무선 인터넷 표준화 포럼 (KWISF)과 한국전자통신연구원 등이 주도하고 투자하여 개발한 모바일 플랫폼으로서 한국정보통신기술협회(TTA) 표준 TTAS.KO-06.0036으로 채택된 바 있다.

앞서 Java, BREW 등에서 설명한 바와 같이 기존의 무선 인터넷 플랫폼은 VM (Virtual Machine), 즉 bytecode²⁾를 사용하는 기술과 binary code를 사용하는 기술로 크게 나눌 수 있다. WIPI는 이러한 대조되는 두 가지 기술의 장점을 동시에 가지도록 개발되어 Java 언어의 장점을 수용하면서 동시에 binary code의 실행 성능을 갖도록 하였다.³⁾ WIPI 플랫폼은 C/C++/Java를 사용

1) 국내 기술로 개발된 모바일 플랫폼 중에는 MAP 외에도 GVM 등이 있으나 이 글에서는 생략하였다.

2) 원래 bytecode라는 용어는 Java에서 사용되는 용어이며 intermediate code가 더 일반적인 용어일 수 있으나 이 글에서는 일반적으로 bytecode라는 용어를 사용한다

3) 이를 위해 J2ME/KVM에서와 같이 finalization 등의 일부 Java feature에 대해서는 제약을 가하였다.

하여 프로그램 개발이 가능하게 하고, 단말기에 탑재되는 WIPI 애플리케이션은 COD(Compile On Demand) 기법에 의해 binary code로 만들어져 다운로드된다. 즉, Java 실행 환경은 classloader, verifier, JIT compiler 등을 포함하게 되는데 COD 게이트웨이에서 AOT compile을 마친 binary code를 단말기에 다운로드 하고 WIPI 단말기에서는 class loader와 유사한 기능을 가지는 binary code loader를 가짐으로써 다운로드된 binary code가 수행될 수 있도록 한다.

WIPI의 구조는 크게 HAL (Handset Adaptation Layer), WIPI Run-time Engine, Basic API, WIPI Application Manager 등으로 나누어진다. (그림 2 참조) HAL은 WIPI 플랫폼이 기반 시스템과 무관하게 동작할 있도록 하는 이식층을 제공한다. Basic API는 WIPI 애플리케이션 개발자를 위해 C/Java 형태의 기본 라이브러리를 제공한다. Application Manager는 애플리케이션의 다운로드, 설치 등과 같은 기본 관리뿐 아니라 API 및 컴포넌트들의 추가/갱신 등의 역할을 수행한다.

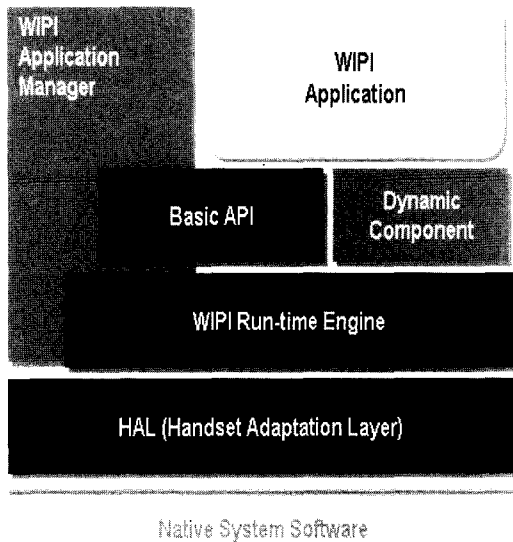


그림 2 WIPI 구조

3. 단말기 제조 기반 모바일 플랫폼

앞 절에서 설명한 무선 인터넷 기반 모바일 플랫폼과는 달리 소프트웨어 구조적으로 보다 하부 계층까지를 포함하는 모바일 플랫폼이며 따라서 일반적으로 플랫폼의 기능이 훨씬 다양하고 보다 뛰어난 CPU의 성능/큰 메모리를 필요로 한다. 예를 들어 Windows Mobile for Smartphone의 경우에는 최소 100MHz 이상으로 동작하는 CPU와 (예: TI OMAP 710, PXA250), 10MB 이상의 코드를 위한 메모리 크기 (예: 32MB ROM/RAM) 등을 요구하므로 고가의 정보 처리용 휴

대폰에 주로 사용된다. 단말기 제조 기반 모바일 플랫폼은 무선 인터넷 기반 모바일 플랫폼과 경쟁 구도에 있다 기보다는 주로 이 플랫폼 위에 무선 인터넷 기반 모바일 플랫폼을 탑재하는 형태를 가진다.

3.1 Windows Mobile for Smartphone

Windows Mobile for Smartphone 플랫폼은 Windows로 PC OS 시장을 장악한 Microsoft의 휴대폰으로의 영역 확장 전략에 따라 추진되고 있는 Windows CE 기반의 플랫폼이다.⁴⁾ 따라서 Windows CE의 주요 기능인 soft real-time scheduling kernel, virtual memory, power management, Object Store, Active Sync, RAPI 등의 기능을 가지고 있으며 Windows Mobile for Pocket PC와의 주요 차이는 펜 입력에 의존하던 방식에서 휴대폰에서 일반적으로 사용되는 소프트 키 입력 방식을 지원한다는 점이다. 지원되는 애플리케이션의 종류는 Pocket Outlook, ActiveSync, Pocket IE, MSN Messenger, Windows Media Player 등 주요 Windows CE 애플리케이션들을 포함한다. 최근에는 COM/DCOM 기반의 component framework에서 나아가 다양한 web service를 지원하기 위한 .NET Compact Framework을 탑재하는 형태로 발전해 가고 있다. 그림 3은 Windows Mobile for Smartphone의 개략적인 구조를 보여준다.]

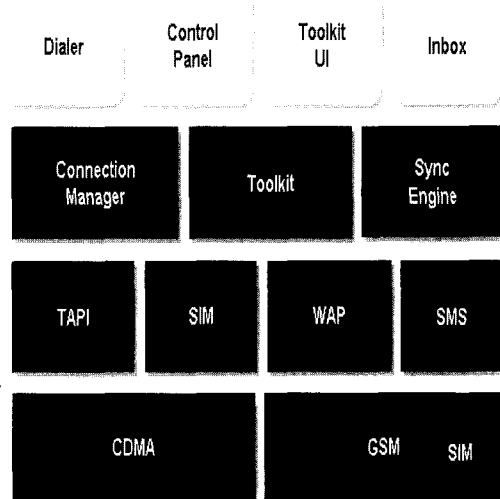


그림 3 Windows Mobile for Smartphone 구조

고성능 휴대폰 시장의 OS 분야를 Microsoft에 잃지 않으려는 기존 모바일 관련 단말기/소프트웨어/사업자들의 많은 견제에 의해 Window Mobile for Smartphone

4) 예전에는 Microsoft SmartPhone 2003으로 불리다가 최근에 Windows Mobile for Smartphone이라는 용어를 사용하게 되었다.

이 탑재된 휴대폰의 시장 점유율은 10%선에 그치고 있고, 현재 이 플랫폼을 채택한 단말기 업체들은 삼성전자, Motorola, Orange 등이 있다.

3.2 Symbian OS

Symbian은 1998년 Nokia, Motorola, Ericsson Matsushita, Psion 등의 회사들이 투자하여 만든 스마트폰 OS 전문 회사이고 이 회사에서 현재까지 개발하고 있는 모바일 플랫폼이 Symbian OS이다. 최근 삼성전자에서 지분 투자를 하여 신문 지상에 오른 적이 있는데 Symbian이 관심을 받는 이유는 Microsoft가 시장 확대를 노리는 고성능 휴대폰 OS 시장에 기술적으로 대응할 수 있는 거의 유일한 솔루션이며 현재까지 스마트폰 OS 시장의 약 50%를 점유하고 있기 때문이다.

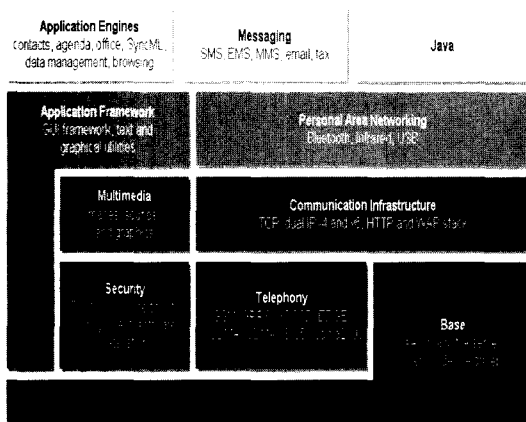


그림 4 Symbian OS 구조

그림 4는 Symbian OS 7.0의 구조를 보여준다. Symbian OS는 크게 Base를 기반으로 Security, Telephony, Multimedia, Communication, Application Framework, PAN, Application Engine, Messaging, Java 등으로 나뉜다. 즉, contacts, schedule, messaging, browsing 등의 다양한 application engine, 녹음, 녹화, 재생, 스트리밍의 A/V 기능, 그래픽 가속 API 등이 제공되는 application framework, IPv4/v6, IrDA, Bluetooth 등의 통신 기능, 3G 기술까지 모두 포함하는 bearer/SIM 기능, 인증서 관리, WIM framework, secure protocol 등의 보안 기능들을 모두 제공한다.

휴대폰은 PC와는 달리 LCD의 크기, 입력 방식, 키 배열 등의 가변성이 크고 제한된 자원을 최대한 효율적으로 사용해야 하므로 광범위한 휴대폰 범위를 동시에 만족시키는 플랫폼을 만드는 것은 매우 어려운 일이다. 이에 따라 Symbian은 공통적인 GUI 컴포넌트인 UiKon GUI 라이브러리 계층까지를 커버하고 실제 UI/GUI와 밀접한 연관이 있는 GUI 라이브러리의 특화 부분은 휴

대폰의 형태와 사양에 따라 구분하도록 하였다. 예를 들어 Nokia는 Series 60, Series 80, UIQ와 같은 Symbian OS 기반의 상위 애플리케이션 플랫폼을 개발하였고, Symbian OS와의 조합에 의해 하나의 제품군이 이루어질 수 있도록 하였다. 예를 들어 Nokia 7650은 Symbian OS와 Series 60을, SonyEricsson P900의 경우 Symbian OS와 UIQ를, Nokia 9290 Communicator는 Symbian OS와 Series 80을 사용한 제품이다.

3.3 PalmOS

PalmOS는 PDA라는 것이 생소하던 시기에 PalmOS 1.x~3.x를 탑재하여 Motorola 68EZ328과 같이 낮은 사양의 CPU에서도 뛰어난 성능, 적은 전력 소모 기술, 그래픽 기술 등을 선보여 전세계 PDA 시장을 석권하였으나 이후 많은 변화를 겪어 작년 초 Palm 소프트웨어 전문 기업인 PalmSource에 의해 개발되고 있다.

PalmOS 1.x~3.x는 Memory, Sound, System, Key Manager, Serial Manager 등의 관리자들에 Shell 역할을 하는 Application Launcher를 탑재하여 PDA용 플랫폼의 형태를 갖추었다. PalmOS 4에서 Telephony API를 보완하면서 스마트폰 OS로서의 변화를 시도하였으나 아직까지는 AMX라는 RTOS 상의 하나의 task로 PalmOS를 동작시키는 이중 구조를 가지는 OS였다. 하지만 PalmOS의 약점인 멀티미디어, 보안 기능 등을 강화하기 위해서 강력하고 보다 범용적인 CPU core를 지원할 필요가 있어 기존 version과는 상당한 차이를 가지는 PalmOS 5를 개발하였다.

PalmOS 5에서는 멀티미디어, 보안, 무선 등의 기능이 대폭 강화되었으며, ARM-compliant CPU를 지원하여 Microsoft, Symbian과의 격차를 줄일 수 있게 되었다. 하지만 기존의 수많은 PalmOS 애플리케이션이 있다는 것이 PalmOS의 가장 큰 장점이므로 이 애플리케이션들과의 호환성을 위해 PACE (Palm Application Compatibility Environment)라는 소프트웨어 에뮬레이션 환경을 제공한다. 나아가 기존의 160x160으로 고정된 LCD에 맞추어진 애플리케이션의 수행을 보장하면서 동시에 고해상도 LCD를 지원할 수 있게 하기 위해 기존 크기에 비례하는 320x320 LCD를 지원할 수 있도록 고해상도 API를 추가하였다. PalmOne은 PalmOS 5를 탑재한 Tungsten W, Treo 600 등으로 미국은 물론 유럽 시장에까지 진출을 시도하고 있다.

3.4 Embedded Linux

Linux는 엔터프라이즈 환경에서 임베디드, 모바일 환

경에 이르기까지 모든 업체가 주목하고 있는 오픈 소스의 첨병이다. 하지만 다른 모바일 플랫폼에 비해 Embedded Linux는 오픈 소스에 의한 여러 장점에도 불구하고 주력 모바일 플랫폼으로 자리잡기에는 시간이 더 필요한 상황이며 IDC의 2006년 경의 고성능 휴대폰 OS 시장 예측에 의하면 Symbian OS, Windows Mobile for Smartphone, PalmOS가 주종을 이룰 것이고 Embedded Linux를 탑재한 휴대폰은 5% 이하에 그칠 것으로 예상하고 있다. 하지만 Embedded Linux를 휴대폰에 탑재하려는 노력은 Motorola 등의 단말기 제조업체와 NTT DoCoMo 같은 사업자 등에서 지속적으로 이루어지고 있으며, 이미 작년 초 Motorola는 Embedded Linux 시장에서 선전하고 있는 MontaVista의 Embedded Linux 기반의 A760 휴대폰을 선보였다.

MontaVista Linux CE Edition을 기준으로 설명하면, Linux kernel에 O(1)의 real-time scheduling 기능을 넣어 preemptive RTOS kernel의 기능을 하도록 하였고, power management, fast boot, XIP (eXecution in Place) 기능 등을 추가하여 내장형 기기에 적합하도록 하였다. 여기에 Qt/Embedded를 사용하여 window system을 갖추고 Qtopia와 같은 애플리케이션 스위트를 보강하여 기존 모바일 플랫폼에 대응하려는 것이 MontaVista의 전략이다.

3.5 Mocha

Mocha (Modular & Configurable Handset S/W Architecture)는 2000년 초부터 삼성전자에서 자체 개발하기 시작한 모바일 플랫폼 및 애플리케이션의 총칭이다.

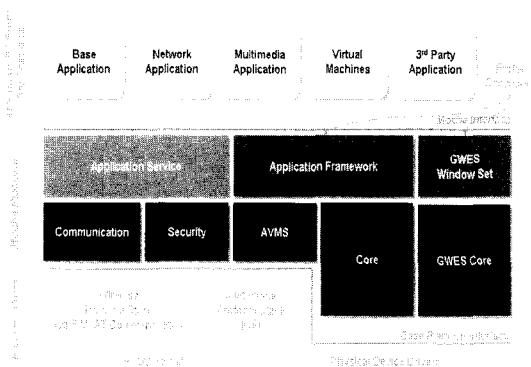


그림 5 Mocha 구조

그림 5는 Mocha의 전체 구조를 보여준다. Mocha는 RTOS kernel, physical device driver, wireless protocol stack (예: CDMA, GSM/GPRS)이 제공되는 환경 위에 이식되며 OS 계층에서 application engine까지를 모두 포함하는 플랫폼이다. 가장 하부에는 Core subsystem이 있는데 여기에는 다양한 RTOS를

지원하기 위한 OS Abstraction Layer, NAND/NOR flash memory를 지원하는 file system, database, DLL Manager 등이 포함된다. GWE (Graphics, Window and Event) subsystem은 primitive graphics operation, window management, event processing 등을 담당하게 되어 Core subsystem과 더불어 전체 플랫폼의 기반을 이룬다. 여기에 Communication, Security, AVMS라는 subsystem이 추가되게 되는데, Communication subsystem은 WAP, TCP/IP, HTTP, Telephony API등을 포함하고 있으며 TAPI의 경우 CDMA/GSM 계열의 2G~3G까지를 모두 포함하는 bearer 독립적 특징을 가진다. Security subsystem은 cryptography, secure protocol, WPKI, DRM 등을 지원한다. AVM (Audio and Visual Multimedia) subsystem은 멀티미디어와 관련된 모든 기능, 즉 A/V media API, image CODEC, H.324M API 등을 지원함으로써 카메라, A/V관련 기능 및 화상 통화 기능 등을 지원한다. 나아가 확장된 2차원, 3차원, vector graphics를 지원하는 엔진이 탑재되어 있다. 플랫폼의 가장 상위에는 Application Service, Application Framework, GWES Window Set이 위치하고 있다. Application Service는 contacts, organizer 등의 애플리케이션 엔진, SyncML, XML parser 등을 포함한다. Application Framework은 Resource, Application, Locale Manager 등을 포함한다. 마지막으로 Window Set은 일종의 UI control로써 휴대폰의 UI/GUI에 따라 달라지는 기능을 모델마다 다르게 적용하기 위해 GWES Core 부분과 분리되어 있다. 이와 같은 플랫폼 구조 위에 3G 고성능 휴대폰군이 가져야 하는 모든 주요 애플리케이션, 즉 PhoneShell, PIMS, Browser, Messenger, Multimedia 애플리케이션 들을 보유하고 있다.

Mocha는 초기 설계부터 효율성/이식성/재구성성 등을 주요 설계 목표로 개발되었기 때문에 그 구성에 따라 다양한 크기와 수의 LCD, 한 개 혹은 그 이상의 CPU 등을 지원할 수 있도록 되어 있고 ARM7 이상의 CPU core에서는 문제없이 수행될 수 있는 성능을 가진다. 현재 Mocha는 삼성전자 내에서 국내 및 해외 시장을 위한 3G 휴대폰 개발에 적극 활용되고 있다.

4. 모바일 애플리케이션 스위트

지금까지 설명한 다양한 모바일 플랫폼들과 모바일 애플리케이션 스위트의 차이점은 플랫폼 주요 모듈의 계층과 API에 있다. 즉 애플리케이션 스위트는 주요 모듈이 browser, messenger, contacts 등의 애플리케이션 엔진이며 WAP stack, SyncML, window system 등은

애플리케이션을 보조하는 공통 서비스 의미를 가지므로 API의 사용자가 일반 애플리케이션이라기 보다는 애플리케이션 엔진 자체가 되게 된다. 이 흐름은 일반적으로 browser를 주력으로 개발한 기존 업체에서 browser를 중심으로 MMS, Instant Messaging SyncML, Java 등의 다양한 애플리케이션으로 영역을 확장/통합해 가면서 자체 window system, security protocol 등을 내장하고 공통적인 porting layer와 UI customization layer를 정의함으로써 생겨난 것으로 점차 모바일 플랫폼의 형태를 갖추어가고 있다.

Openwave의 경우 System Programming Interface 아래 모바일 플랫폼의 공통적 기능 (예 : network storage, security, media/graphics services)을 두고 Application Programming Interface에는 애플리케이션 용 컴포넌트를 배치하여 browser MMS, phonebook, SyncML 등을 포함하도록 하였다. Teleca Obigo의 경우 Obigo framework를 만들어 작은 모바일 플랫폼 기반을 구성한 뒤 여기에 기존에 제공하던 browser messenger, Java application manager, security 기능 등을 추가하여 애플리케이션 스위트를 구성하였다. Nokia TSS (Terminal S/W Suite)는 임의의 휴대폰에 이식 가능하도록 성능/크기/이식성을 중점으로 개발하였기 때문에 Symbian OS 기반의 Series 60 등과 차별화되며 TSS에는 Browser, MMS, SyncML, DRM Email, LBS 등이 포함된다.

5. 결 론

이 글에서는 광범위하게 사용되는 모바일 플랫폼이라는 용어를 크게 무선 인터넷 기반 모바일 플랫폼, 단말기 제조 기반 모바일 플랫폼, 모바일 애플리케이션 스위트, 세가지로 구분하여 각 분야의 주요 플랫폼의 구조와 기능에 대하여 개략적으로 설명하였다.

금년에도 모바일 기술의 발전과 시장의 성장은 계속될 것으로 예상되며 이 흐름의 최전선에 모바일 플랫폼이 있는 만큼 각 사업자, 단말기 제조사, 플랫폼/애플리케이션 소프트웨어 개발사 등의 경쟁도 더욱 치열해질 전망이다. 기술적으로는 3G로 점차 기술 중심이 이동하면서 국내에서도 Smart Card를 탑재한 휴대폰이 늘어날 것이고 A/V 멀티미디어에 대한 요구가 증대되면서 메모리 카드, 3D, vector graphics 등의 적용이 휴대폰의 기능과 UI에 많은 변화를 가져올 것으로 예상된다. 그리고 IPv6, Instant Messaging & Presence, Super Distribution 기능 등이 적용된 제품이 출시될 것으로

보여 이와 같은 모바일 기술 변화에 적절히 대응한 모바일 플랫폼의 성장세가 예상된다.

무선 인터넷 기반 플랫폼에 있어서는 Java는 안정적인 성장세를 이룰 것으로 보이며 앞서 설명한 바와 같이 새로운 API의 표준화 및 상품화가 이어질 것이다. QUALCOMM은 BREW, Launchpad를 앞세워 CDMA chipset과 protocol에 이어 애플리케이션 분야로 수익을 확장하려고 하고 있고 WIPI over BREW 등으로 국내 시장의 재진입을 서두르고 있어 WIPI 업체의 수성이 주목된다. 단말기 제조 기반 플랫폼에 있어서는 .NET Compact Framework 을 포함한 Microsoft의 Windows Mobile 전략이 Nokia 등의 견제를 넘어 어느 정도 시장에 영향을 줄 수 있을지 주목된다. 반면 Palm OS의 경우는 CDMA시장과는 달리 GSM시장에서는 그리 성공적인 확산을 하기는 어려울 것으로 보이며 Embedded Linux를 탑재한 휴대폰은 실험적으로나마 계속 여러 제조 업체를 통해 상품화될 것으로 예상된다.

참고문헌

- [1] Java 2 Micro Edition, <http://java.sun.com/j2me/index.jsp>
- [2] QUALCOMM BREW, <http://www.qualcomm.com/brew>
- [3] MAP, <http://www.mobiletop.co.kr>
- [4] 한국무선인터넷표준화 포럼, <http://www.kwisforum.org>
- [5] Windows Mobile, <http://www.microsoft.com/windowsmobile>
- [6] Symbian OS, <http://www.symbian.com>
- [7] PalmOS, <http://www.palmsource.com/pamos>
- [8] CE Linux Forum, <http://www.celinuxforum.org>
- [9] MontaVista, <http://www.mvista.com>
- [10] EMP, <http://www.ericsson.com/mobileplatforms>
- [11] Openwave, <http://www.openwave.com>
- [12] Teleca Obigo, <http://www.obigo.com>
- [13] Nokia TSS, http://www.nokia.com/downloads/solutions/mobile_software/NTSS_product_brief.pdf
- [14] Magic4, <http://www.magic4.com>
- [15] 배석희, '모바일 플랫폼 표준화 동향 및 향후 발전 방향', TTA 저널, 제 82호, 2002. 20p

홍 준 성



1991 고려대학교 전자전산공학과 학사
1993 한국과학기술원 전기및전자공학과 석사
1998 한국과학기술원 전기및전자공학과 박사
1998~현재 삼성전자 CTO 전략실 소프트웨어센터 수석
관심분야 : Mobile S/W platform, Mobile application, S/W architecture, Software engineering, Distributed computing
E-mail : jshong@samsung.com

• **The 6th Korean Conference on Software Engineering (KCSE 2004)** •

- 일 자 : 2004년 2월 19 ~ 21 일
- 장 소 : 강원도 휘닉스파크
- 주 최 : 소프트웨어공학연구회
- 상세안내 : <http://se.dongguk.ac.kr>