

모바일 단말기용 퍼스널 자바 플랫폼에 대한 연구

권오형^o, 진민식, 유흥식, 조증보, 정준영, 정민수
경남대학교 컴퓨터공학과

Research of PersonalJava Virtual Machine Platform for Mobile Device

OhHyung Kweon, MinSik Jin, Hong Sik Yoo, JeungBo Cho, JunYung Jung, MinSu Jung
Dept. of Computer Science, Kyungnam Univ.
E-mail : opus@hawk.com.kyungnam.ac.kr

요 약

자바는 썬 마이크로 시스템사가 1999년 6월에 플랫폼이나 개발 성격에 따라서 3가지 Edition으로 나누었다. Java 1.1 버전까지는 일반적인 자바 Applet 이나 Application 의 개발이 목적이었다. 그러나 Java 1.2 버전 이상에서는 자바의 용도가 좀 더 다양해지고 자바가 올라갈 수 있는 플랫폼이 다양해져 그 특성에 맞는 자바가 등장하게 되었다. 본 연구는 모바일 단말기용 Java VM 플랫폼 중 PVM 플랫폼을 연구하는 데에 기초로 한다.

1. 서론

일상 생활에서 자주 접하게 되는 컴퓨터와 여러 가지 전자 제품, TV, VCR, 오디오, 이동전화 등에 마이크로프로세서가 장착되고, 또한 이것들이 네트워크로 연결되는 변화는 새로운 소프트웨어의 필요성을 요구하기 시작했고, 이러한 환경을 목표로 나온 하나의 도구가 바로 자바 기술이다. 자바(Java)는 플랫폼 독립적인 특성으로 인하여 인터넷과 같은 네트워크 환경에서 동작하는 프로그램 작성에 널리 사용되고 있다.

퍼스널 자바는 3년 전 제한된 리소스를 가진 소형 기기에 적합하도록 썬 마이크로 시스템사에서 임

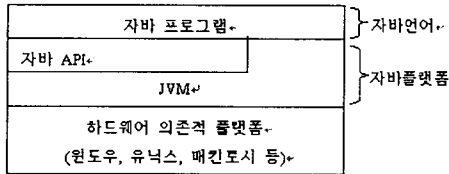
베디드 자바, 자바 카드와 더불어 이 세가지 기술들을 소개했었다. 즉 퍼스널 자바(Personal Java)는 임베디드 자바(Embedded Java)와 더불어 지금 현재 커다란 관심을 갖고 있는 J2ME의 모체적 성격을 지닌 Java스펙이다. 이 기술 중 퍼스널 자바는 개인용 내장형 장비에서 수행되는 네트워크 응용프로그램을 위한 자바 애플리케이션 환경이다. 이는 개방형 시스템 운용기술로 기존의 데스크탑 컴퓨터보다는 스크린 폰, PDA, 웹TV, 하이엔드 디지털 셋톱박스, 하이엔드 스마트폰 및 웹폰, 영상전화 용도 등 소비자 정보 가전 제품에 맞는 애플리케이션에 활용할 수 있다. 네트워크에 연결할 수 있는 소형 기기에 적합한 소규모의 Java 실행 환경을 정의한 것으로

ClassicVM 기반으로 구현되는 것을 전제하고 있다.

2. 관련연구

2.1 자바의 구성요소

자바를 언급하면 일반적으로 자바 언어 또는 자바 프로그램을 떠올린다. 그러나 컴퓨터가 하드웨어와 소프트웨어로 구성돼 있듯이, 자바 또한 크게 두 가지로 나눌 수 있다.



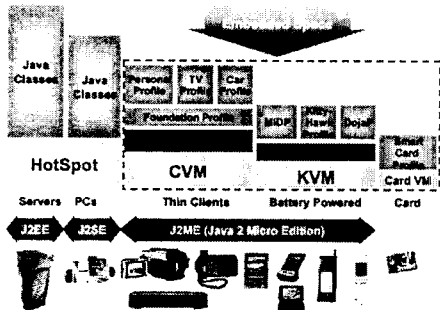
[그림 1] 자바의 구성요소

자바언어 : JVM에서 실행하기 위한 프로그램을 작성할 수 있도록 해주는 자바 프로그래밍 언어를 이용하여 자바 프로그램을 작성한다.

자바플랫폼 : 자바프로그램을 실행시켜주기 위한 자바가상머신(Java Virtual Machine)과 자바가상머신에서 프로그램을 실행할 때 사용하는 라이브러리를 가리켜 자바플랫폼 이라 한다.

2.2 자바플랫폼과 JVM 종류

JVM은 JVM이 포팅되는 자바 플랫폼이나 동작하는 시스템의 성능, 디바이스의 특성에 맞춘 API라이브러리를 기준으로 자바 2플랫폼을 분리하였다.



[그림 2] JAVA2 플랫폼

• J2EE 플랫폼과 기술

서버측면의 멀티어 엔터프라이즈 애플리케이션의 개발과 배치를실현하는 J2ME기반의 플랫폼이다 기업 환경에 맞는 자바 기술이 들어가게 된다. JSP, 서블릿, EJB, JNDI, 자바 IDL & CORBA, RMI over IIOP, JDBC, 메시지 서비스, 자바메일 등.

• J2SE 플랫폼과 자바기술

자바언어를 이용한 애플리케이션, 애플릿, 컴포넌트 의개발 및 운영환경이다. 기본적으로 자바로 PC나 Workstation 기반에서 자바가 할 수 있는 모든 일을 지원한다.

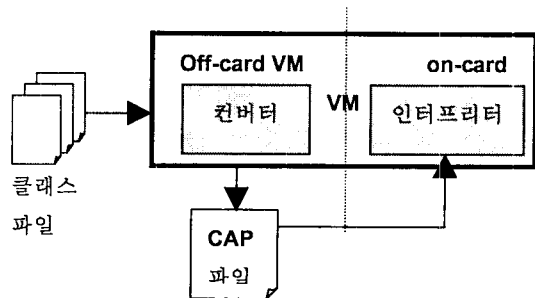
자바 핫스팟 성능 엔진, 자바2 보안 모델, 자바 플러그인, 자바빈즈 컴포넌트 API, 자바 IDL

• J2ME 플랫폼

임베디드 디바이스의 다양하고 제한된 환경에 자바를 탑재하기 위한 플랫폼이다. 적은 용량의 VM과 기본적인 자바 API 보다 많이 절약된 API가 올라간다. 핸드-헬드 디바이스(Hand-Held Device)나 PDA, 스크린 폰(Screen Phone), 셋톱박스(Set-top Box), 넷TV(net TV)와 같은 디지털 기기들이 네트워크로 연결되어 있고,

• 자바카드 플랫폼

자바카드 플랫폼은 자바언어와 자바카드 가상기계를 기반으로 하여 내장형 시스템 중에서 메모리, 계산 처리능력이 가장 작은 규모에 속하는 스마트 카드에 적용되는 시스템으로 기존의 자바 플랫폼에 비해 많은 제약 사항과 수정 및 추가 사항을 내포하고 있다



[그림 3] 자바카드 가상머신 구조

3. 퍼스널자바(Personal Java)

3.1 컨피규레이션과 프로파일

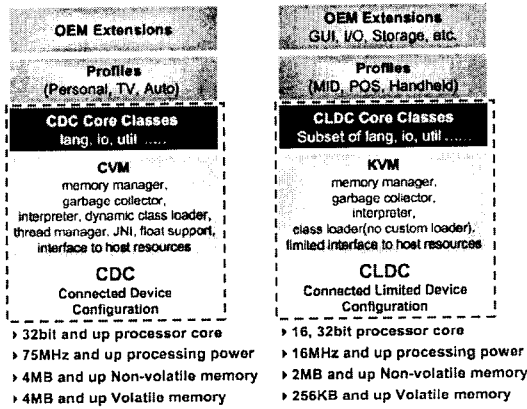
컨피규레이션(Configuration)이란 자바 가상 기계와 코어 API들에 대한 명세를 의미한다. 메모리와 CPU 등의 크기와 성능이라는 측면에서의 요구사항이 동일한 디바이스들의 집합을 하나로 묶어서 컨피규레이션을 정의한다. J2ME는 다중 컨피규레이션을 수용하고 있으며, 특히 하드웨어 메모리 사양이 128-512KB인지, 512KB 이상인지 대한 관점으로 다음과 같은 두 가지의 컨피규레이션을 정의하고 있다.

CLDC

K-자바 가상 기계 기반의 이동, 개인, 접속 디바이스들을 위한 컨피규레이션이다

CDC

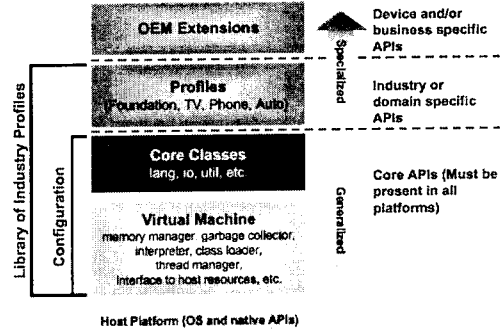
CDC는 기존의 퍼스널 자바(Personal Java)를 바탕으로 프로세서는 32비트에서 동작하고 CLDC보다 많은 리소스를 가진 디바이스를 이한 컨피규레이션이다.



[그림 4] J2ME Configuration

CLDC 는 KVM 을 사용하고 CDC 는 PersonalJava VM 이나 Classic VM 을 사용한다

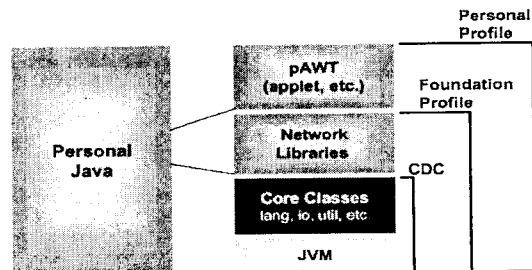
프로파일이란 특정한 각 디바이스들의 기능, 혹은 버티컬 시장의 요구사항에 맞추어 CLDC, CDC 컨피규레이션을 기반으로 필요한 클래스라이브러리에 대한 명세이다.



[그림 5] 프로파일 구조

3.2 퍼스널 자바

퍼스널 자바는 CDC(Connected Device Configuration)를 기반으로 하여 기존의 퍼스널 자바(Personal Java)를 바탕으로 자바 플랫폼을 위해 512KB(주로 2MB) 이상의 메모리가 사용 가능한 디바이스들에 적용되도록 설계 된 매우 콤팩트한 형태로 50MHz급의 32비트 CPU 이상에서 동작된다. 퍼스널 자바는 기존의 자바 API를 모두 갖추고 있고 부분 집합으로 구성되어 있지만, 서버에서 필요한 라이브러리를 다운 받는 방식으로 거의 모든 자바 애플릿을 실행시킬 수 있다. 이는 퍼스널 자바 스펙에 따라 작성된 애플릿은 JDK1.0.2와 JDK1.1로 작성된 애플릿 환경에서도 수행이 된다는 것이다. 그리고 원격제어, TV 출력, 터치 스크린과 같은 가전 제품의 다양한 입출력 포맷을 다룰 수 있을 정도의 유연성을 제공해야 하는데 이런 경우는 특정 장치의 입출력 포맷을 지원하기 위한 클래스 라이브러리가 퍼스널자바 API에는 없기 때문에 실제 퍼스널 자바를 타겟 시스템에 적용시키려면 별도의 클래스 라이브러리 즉, java.io 패키지를 확장하여 사용해야 한다.



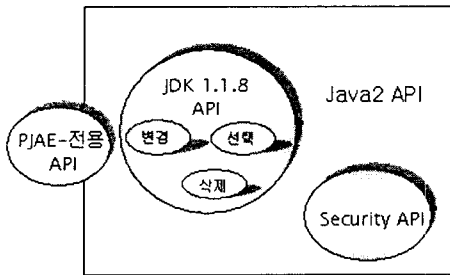
[그림 6] 퍼스널 자바 구성요소

그림과 같이 퍼스널 자바 구성요소는 CDC + Personal Profile + Foundation Profile 로 이루어진다.

CDC는 JVM(CVM- C Virtual machine)과 최소한의 클래스 라이브러리와 API들을 포함하고 있다. 기기상에서 실행하기 위해서는 프로파일이 필요하다. CDC를 위한 세가지의 프로파일 영역이 제안되었는데 Personal Profile, Foundation Profile, RMI Profile 이 있다. 파운데이션 프로파일은 두가지의 목적이 있는데 하나는 네트워크를 필요로 하는 기기를 위한 API를 지원하는 것이고 다른 하나는 다른 프로파일들을 위해 기반 프로파일 역할을 한다. 따라서 파운데이션 프로파일은 다른 프로파일과 연관되어 존재할 수도 있고 단독으로 존재할 수도 있다. 퍼스널 프로파일은 인터넷 연결을 위해 필요한 그래픽과 네트워크를 지원한다.

3.3 PJAE(PersonalJava Application Environment)

퍼스널 자바는 기존의 JDK1.1.8 API의 패키지, 메소드, 클래스를 수정하여 정의한 것으로서 보안 기능은 Java2 Security API를 일부 채택하였다. 즉, PJAE 내의 API들은 JDK1.1.8을 기반으로 필수API, 선택API, 미지원API, 전용 API로 나눈다.



[그림 7] PJAE 개념도

그림과 같이 기본적으로 퍼스널자바는 JAVA2 API의 상당 부분을 수용하고 있으며 동근 부분이 퍼스널 자바에서 수용하고자 하는 범위이다.

- 변경 API : 기존의 API에서 변경되어 새로 정의한 API
- 미지원 API : 기존의 API에서는 정의되어 있으나 PJAE에서는 지원되지 않고 삭제된 API

- 전용 API : 퍼스널 자바에서만 사용 가능하고 기존의 JDK에는 없는 API
- 선택 API : 구현자의 선택으로 인해 API를 지원할 수 있는 API
- 보안 API : JAVA2 의 Security API를 사용

<표 1> PJAE API 의 대표적 요소

필수	변경	선택	전용
java.applet	java.awt	java.math	com.sun.lang
java.beans	java.io	java.rmi	com.sun.util
java.text	java.lang	java.security	.Ptimer
	java.lang.reflect	java.util	java.awt.Component
	java.security	java.net	ponent.isDoubleBuffered

4. 퍼스널자바 소스 설치 및 컴파일

퍼스널 자바 소스를 압축 풀면 build, src 두개의 중요한 서브 디렉토리를 생성한다 여기에는 바이너리 파일과 APIs 그리고 KVM source 파일과 API 소스 파일들이 포함 되어있다.

- build : 목표 개발 환경상에 PJAE 만들기 위한 스크립트와 make 파일들을 포함하고 있다.
- src : PJES에 대한 소스코드를 포함하고 있다.

• Personal Java Build 소프트웨어 요구사항

- JDK 1.1.8, Microsoft Visual C++ (6.1)
- Microsoft Macro Assembler (6.11)
- MKS Toolkit (6.1), GNU Make (3.77)
- Windows NT Workstation (4.0)

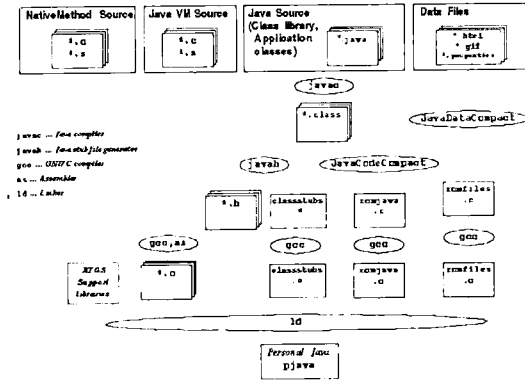
1. 소프트웨어를 설치하고 환경변수를 설정한다.
2. defs.win32.mk 의 파일을 자신의 시스템에 맞게 수정한다.

```
JAVATOP = d:/JDK1.1.8, ROMJAVACFLAGS = /Zm1500
```

3. MKS ToolKit을 실행시켜 make 파일을 실행시킨다.

4. 컴파일이 완료되면 pjava_g.exe , pjavaw_g.exe pappletviewer_g.exe 이 생성

된다



[그림 8] 퍼스널 자바 컴파일 과정
PersonJava의 소스파일이 컴파일 되어 퍼스널 자바 pjava_g.exe 를 생성하는 단계를 나타낸 그림이다

5. 퍼스널자바 실행

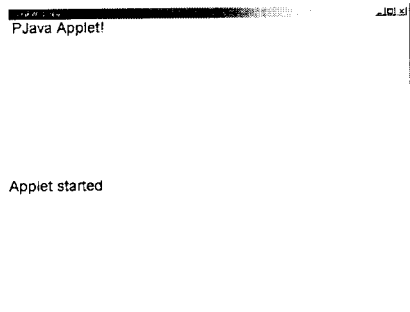
본 논문에서는 컴파일 하고 난 결과인 pjava_g.exe, pjavaw_g.exe, pappletviewer_g.exe 중에서 애플릿 프로그램을 실행시키기 위해 pappletviewer_g.exe를 사용하였다. 일반 애플리케이션은 pjava_g를 사용.

자바 프로그램을 작성한다. (HelloApplet.java)

컴파일러는 기존의 자바 컴파일러를 통해 프로그램을 컴파일한다. (javac HelloApplet.java)

컴파일 결과 HelloApplet.class 파일이 생성된다.

컴파일한 클래스 파일을 pappletviewer_g.exe 를 사용하여 실행시킨다. (pappletviewer_g http://vinus241.com.kyungnam.ac.kr/main/test.html)



[그림 9] 실행결과

6. 결론 및 향후 연구 과제

본 논문에서 제시된 퍼스널 자바 가상 기계가 KVM에서 제공하지 않았던 많은 API관련된 것을 보충함으로써 KVM의 문제점이었던 JINI와의 연동을 통해 모바일 단말기에서 보다 광범위의 서비스를 제공함과 더불어 모바일 기기의 하드웨어 사양이 높아지고 퍼스널 자바가 타겟으로 하고 있는 거대한 PDA 시장의 성장에 힘입은 애플리케이션 개발과 무선기기의 기술적인 발전으로 인해 앞으로는 퍼스널 자바가 내장되어 기존의 K-자바 가상 기계 보다 향상된 성능을 제공할 것으로 기대한다.

향후 연구 과제로는 본 논문에서 연구한 VM을 바탕으로 실제 모바일 단말기상에 가상머신 포팅에 대한 연구 방향을 맞추고자 한다

[참고 문헌]

- [1] E. Giguere, Java 2 Micro Edition, 2000
- [2] <http://java.sun.com/>, Sun Microsystems, Java Home Page
- [3] <http://java.sun.com/products/j2me> Java 2 Platform, Micro Edition
- [4] <http://java.sun.com/products/cdc/>, CDC
- [5] B. Venner, *Inside the Java Virtual Machine*, McGraw-Hill, 1998
- [6] <http://java.sun.com/products/personaljava> PersonalJava
- [7] PersonalJava™ Specification v 1.1.3
- [8] PersonalJava™ Technology White Paper
- [9] PersonalJava™ Environment Software Documentation
- [10] <http://java.sun.com/products/cldc>, CLDC
- [11] <http://www.kvmworld.com>, Micro Java Network
- [12] <http://java.sun.com/products/midp>, MIDP Homepage
- [13] <http://java.sun.com/docs/books/vmspec/index.html> Lindholm, Tim, and Frank Yellin, *The Java Virtual Machine Specification*, Addison-Wesley Pub Co
- [14] <http://www.kvmworld.com>, Micro Java Network
- [15] <http://www.mobilejava.co.kr/>, Mobile Java Developer community