

MIDP에 물레이터를 이용한 미들렛슈트의 구현 사례에 관한 연구

우원택 *

(Won-Taek Woo)

요약 본 논문은 MIDP에 물레이터를 이용한 미들렛슈트의 전 개발 과정을 소개함으로써 개발 틀 독립적인 모바일프로그램의 개발과정을 이해하고 미들렛, Manifest, JAR, JAD, 미들렛 슈트의 각 단계별 개발 스트림의 사례를 예시함으로써 DOS환경에서의 미들렛 슈트의 개발과정을 완전히 숙지하여 NetBeans 모빌리티팩이나 Eclipse와 같은 통합개발환경이나 자바 Wireless Toolkit과 같은 프로젝트형 개발환경에서 자바모바일프로그램의 개발을 보다 용이하게 이해하고자 함에 그 목적이 있다. 최근 자바 모바일프로그램의 개발과 관련된 연구와 문헌들이 많이 제공되고 있으나 기존의 연구들과는 달리 본 논문에서는 미들렛슈트의 개발스트림을 각 단계별로 구현하였음에 의의가 있다고 하겠다.

핵심주제어 : 미들렛, 미들렛슈트, MIDP, J2ME

Key Words : MIDlet, MIDletSuite, MIDP, J2ME

I. 서론

자바를 이용한 모바일콘텐츠의 구현 사례는 다수가 있다. 일반적으로 이들의 구현은 제일 밑단에서 MIDP에 물레이터를 이용하여 구현할 수가 있고 그 다음으로는 Java Wireless Toolkit(JWT) 이라는 툴 키트를 이용할 수가 있으며 마지막으로 NetBeans 모빌리티 팩이나 Eclipse 모바일프로그 인과 같은 통합개발환경을 이용하여 개발할 수가 있다. IDE의 통합개발환경을 이용하면 에러수정이나 프로그램편집을 쉽게 구현할 수 있는 반면에 프로그램 설정관계나 에물레이터의 연결등과 같은 초기 구동을 위한 준비 작업이 다소 복잡한 편이며 JWT를 이용하면 프로그램의 컴파일이나 빌드 작업을 간단히 수행할 수 있는 반면에 프로그램 수정을 위한 에디터가 내장되어 있지 않기 때문에 편집 기능이 다소 미약하다. MIDP를 이용하여 모바일프로그램을 개발하면 프로그램의 구조나 연결 관계를 잘 이해할 수가 있으나 프로그램 설정과정이나 명령어 입력이 다소 복잡하며

편집기능도 없기 때문에 다른 편집기를 이용하여야 한다. 그리고 프로그램의 실행을 위한 명령어들이나 옵션 등이 복잡하여 틀리기가 쉽고 실행이 잘 되지 않을 때도 있다. 그러나 MIDP의 개별개발환경이나 JWT의 프로젝트형 개발환경이나 NetBeans Mobility Pack의 통합개발환경이나 이들 모두에서 생성된 미들렛 클래스파일이나 JAD파일 등을 각각의 개발환경에서 상호 교환하여 사용할 수가 있으며 그렇게 하기 위해서는 DOS환경에서의 수동적 개별 생성 과정과 각각의 개발 환경별 디렉터리 구조와 파일 특성에 대해서 충분한 지식이 요구된다. 따라서 본 사례연구에서는 이들을 위한 시작단계로 MIDP에 물레이터를 이용한 미들렛슈트의 전 개발 과정을 단계별로 소개함으로써 개발 틀 독립적인 모바일프로그램의 개발과 이용을 도모하고자 함에 그 목적이 있다.

II. 문헌연구

2-1. MIDP의 개황

* 대구한의대학교 문화.정보대학 모바일콘텐츠학부

Ridley(2007)에 의하면 2007년 현재 전 세계 인

구의 약 반인 32억 5천만이 모바일 폰을 사용하고 있으며 Ritter(2008)의 조사에 따르면 2008년 현재 25억 개의 자바 폰이 유통되고 있는 것으로 알려지고 있다. Ritter에 따르면 자바기능장치는 60억개이며 35억개의 자바카드, 20백만의 자바셋톱박스, 800백만 개의 자바데스크탑, 180개의 자바콘텐츠배급자, 6백만 개발자가 전 세계에 존재하고 있다고 한다. 이창진(2003)에 따르면 1991년 그린프로젝트로 시작된 자바가 퍼스널자바(Personal Java)를 거쳐 컨피규레이션과 프로파일이 조합된 MIDP 1.0이 개발되어 2001년 봄 MIDP NG로 2002년 겨울에 MIDP2.0 으로 개명되어 최종 발표되었고 현재 MIDP 3.0이 개발 중에 있다고 한다. 하동욱(2003)에 의하면 MIDP 2.0은 기존의 MIDP 1.0의 기능을 강화하고 모바일 커머스로 도약하는 기치를 내걸고 개발되었다. MIDP 2.0의 스펙은 <그림 1>과 같으며 CLDC위에 탑재된다.

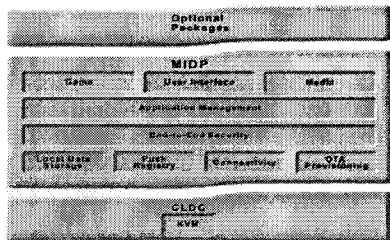


그림 1 MIDP 2.0 스펙

MIDP의 모든 클래스와 CLDC의 모든 클래스는 선 마이크로시스템(2009)의 사이트에서 참조할 수 있으며 이들을 위해 자바커뮤니티프로세스(2009)에서 규정한 자바스펙요구서(JSR: Java Specification Request)의 내역을 보려면 각각 JSR 118과 JSR 139를 참조하면 된다. MIDP 3.0의 스펙리드인 Jim Van Peursesem에 의하면 MIDP 2.0은 XML관련 기술의 부재, 너무 많은 전문가 그룹의 참여 등의 문제가 있으며 자바언어와 VM구조의 변화를 수용할 수 있게 새롭게 개발 중에 있다고 한다. 오늘날 자바 모바일프로그램의 가장 큰 문제점 중의 하나로, 이상적으로는 하나의 미들렛은 어떤 장치에서도 실행 가능하여야하나 실제로는 여러 가지 문제점이 발생하고 있다는 것이다. 이는 작은 모바일 장치나 스크린에 맞추기 위해 시도한 단편화의 문제로 비롯된 것이라 한다. 단편화의 문제는 장치수준단편화(Device level fragmentation), 표준단편화(Standards fragmentation), 실행단편화(Implementation fragmentation)의 세 가지 유형이 있다. 향

후세대의 JavaME 표준은 이러한 단편화의 문제를 최소화 하는데 초점이 주어진다고 한다. MIDP 3.0은 멀티태스킹, 미들렛 간의 커뮤니케이션, 보안과 접근통제 등의 기능이 향상되고 MIDP3.0위에 탑재될 MSA(Mobile Services Architecture)는 On-device 서비스프로그램 제공과 LWUIT(Light Weight User Interface Toolkit)의 사용자인터페이스 툴킷을 제공하여 3D와 SVG를 통합하고 터치스크린을 지원한다. MIDP 3.0은 차세대 광 저장장치인 Blu-ray를 지원하며 이를 이용하면 DVD를 대체할 1084픽셀, 24프레임, 50GB 배정도, 200GB 멀티레이어 저장장치, 고화질무비, 멀티채널(7.1)오디오 기능을 제공할 수 있으며, 디스크 또는 온라인 겸용의 게임플레이도 가능하다고 한다. 아울러 <그림 2>에서와 같은 5개의 독립형 고화질 그래픽평판, 2개의 비디오 평판을 이용하여 배경프레임이 정확한 애니메이션을 제공할 수가 있다.

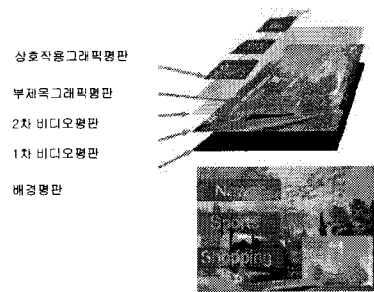


그림 2 5개의 그래픽 평판과 2개의 비디오 평판

정확한 프레임변환 애니메이션은 gfx를 비디오에 동기화하고, 느슨한 비디오싱크, 정확한비디오싱크 등을 통해 달성될 수 있다., <그림 3>는 J2ME 플랫폼의 진화와 확장성을 보여주고 있다.

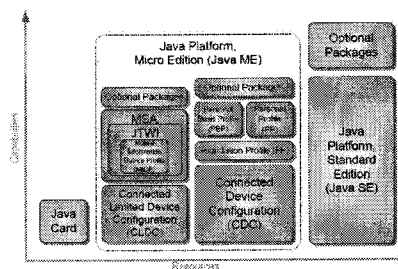


그림 3 플랫폼의 확장과 진화

MIDP에 기반한 J2ME에 관한 연구로는 박기정, 서성채, 김병기(2008), 박영수와 장영철(2004)의 XML관련 미들렛 응용프로그램에 대한 연구가 있

으며, 모바일데이터베이스에 관한 연구로는 김형훈(2009)과 Lu et al(2004)의 연구가 있고, 모바일3D나 3D엔진에 관한 연구로 조종근, 박윤희, 김종민(2005), 주현식, 이송원, 홍성수(2008), 조종근(2008)의 연구가 있다. 모바일보안에 관한 연구로는 박남제와 송유진(2002)과 Itani et al(2004)의 모바일전자지갑과 m-Commerce 보안에 관한 연구가 있다. 최근의 아이폰과 안드로이드폰과 관련한 모바일 애플리케이션 개발에 관한 연구로는 김덕진(2009), 김동현(2009), 김석우(2009), 박성서(2009), 이창진(2009)등의 연구가 있다.

2-2. MIDP 에뮬레이터

MIDP 에뮬레이터는 사용자에게 MIDP 응용프로그램을 데모하기위해 사용되기도 하지만 통상은 개발자들이 응용프로그램을 테스트 하는데 사용한다. 에뮬레이터는 개발 킷이나 패키지의 일부로서 배포된다. 에뮬레이터의 어려운 점은 정확하게 장치의 행위를 흉내 낼 수 있느냐 하는데 그 초점이 있다. 에뮬레이터의 정확성은 개발상의 난이성과 관련되어 있으며 에뮬레이터 개발자들은 자신들의 시간과 기술에 따라 에뮬레이터의 정확도를 결정하여야 한다. 가지각색의 에뮬레이터 중에서 벤더(Vendor)들은 JWT의 에뮬레이터를 사용할 수도 있고 아니면 실제의 프로세스나 장치상의 칩들을 에뮬레이트하기 위해 코드를 작성하기도 한다. 이 경우에는 에뮬레이터는 똑같은 OS-수준의 소프트웨어나 실제장치와 같은 MIDP 실행바이너리들을 실행시킬 수가 있다. 이 방법은 좀 더 어렵고 비용이 더 드는 방법이나 마지막 결과는 실제장치와 훨씬 더 가깝게 에뮬레이터 된다. 에뮬레이터의 종류는 개념에뮬레이터(Concept emulator)와 실세계에뮬레이터(Real life emulator)가 있으며 개념에뮬레이터는 특정장치를 뜻하는 것이 아니고 어떤 종류의 장치들의 일반적 특성을 표시한다. 예를 들면 J2ME JWT은 다양한 MIDP 장치들을 표현하는데 사용되는 개념에뮬레이터이다. 실세계에뮬레이터는 실제장치의 외양과 행위를 흉내 내도록 설계되었다. 이와 같은 에뮬레이터는 장치 그 자체에서 실행되는 모든 바이너리코드나 일부분을 실행한다. Nokia는 두 가지 유형의 에뮬레이터를 모두 제공한다. 회사는 출시될 장치시리즈 들을 준비하는 개념에뮬레이터를 제공하고 장치시리즈가 출시된 후에는 실세계에뮬레이터를 제공한다.

J2ME JWT의 에뮬레이터는 다양한 스킨을 입혀서 여러 가지 외관의 에뮬레이터로 표현할 수 있다. JWT에뮬레이터는 어떤 장치의 외양뿐만 아니라 버튼의 행위나 다른 컨트롤러들을 나타낼 수도 있다. 스크린크기, 컬러깊이, 폰트나 장치의 입력메소드 등을 지정할 수도 있다. JWT의 에뮬레이터는 특정 벤더의 API를 사용하지 않는 응용프로그램을 테스트할 수 있는 범용성의 환경을 제공한다. JWT 에뮬레이터는 윈도우, 리눅스, 솔라리스 모두에서 사용 가능하다. JWT는 성능시뮬레이션, 메소드프로파일링(Method profiling), 네트워크 모니터링(Network monitoring) 등의 유용한 툴들을 개발자들에게 제공한다. 에뮬레이터의 사용에 대한 규칙은 없으나 통상의 에뮬레이터는 JWT나 통합개발환경(IDE)에 팩으로 추가되어 제공된다. 혹은 테스트 툴이나 독립 개발 키트의 한 부분으로 패키지 되어 제공된다. 본 연구에서는 MIDP 2.0에 포함되어 있는 MIDP에뮬레이터를 사용하였다.

2-3. MIDP 프로그래밍

MIDP는 CLDC의 위에서 실행되며 MIDP에서 사용되는 애플리케이션을 MIDlet이라고 하며 하나의 MIDlet은 하나의 클래스 파일로 구성된다. Sun Microsystems(2009)의 MIDP2.0에서 제공하는 API는 사용자인터페이스, 게임, 데이터저장검색, 미들렛, 네트워킹, 미디어, 공용키, 입출력, 자바언어, 유틸리티에 관한 클래스파일들이 패키지로 제공되고 있다. 이중에서도 javax.microedition.midlet에서 제공하는 미들렛 API 클래스에서는 MIDlet 애플리케이션을 시작, 중지, 제거하는 메소드를 제공하며 javax.microedition.lcdui에서는 유저인터페이스를 만들기 위해 필요한 클래스를 제공한다. 애플리케이션에서 사용되는 데이터를 저장할 수 있도록 데이터베이스 기능을 javax.microedition.rms 에서 제공하며 javax.microedition.io의 HTTPConnection에서는 http 커넥션을 위한 네트워킹을 제공한다. java.util.Timer 나 java.util.TimerTask에서는 애플리케이션이 일정시간 간격으로 작업을 수행할 수 있도록 예약기능도 제공한다. MIDP프로그래밍을 하기 위해서는 JDK1.6을 Sun Microsystems(2009)의 사이트에서 다운로드받아 설치해야한다. JDK를 설치한 후 실행파일을 path로 잡아주고 클래스파일도 classpath로 잡아주는 환경 설정을 하여야 JDK 설치가 완료된다. JDK 설치 다음으로 midp-fcs를

설치해야 되는데 여기에도 실행파일과 클래스 파일이 있으므로 JDK와 같이 path와 classpath의 환경을 설정해 주어야 한다. 이와 같이 MIDP프로그래밍을 위한 소프트웨어 툴킷이 설치되고 나면 J2ME로 미들렛 자바프로그램을 작성한 후 미들렛의 구성을 설명해 주는 manifest파일을 작성하고 난 후 이들 두 파일들을 JAR파일로 압축하고 JAR파일의 내용을 설명해 주는 JAD파일을 만든 후 MIDP에플래이어에 JAR와 JAD파일을 인스톨하여 실행시키면 MIDP프로그램의 구현이 완성된다. 이때 JAR와 JAD를 인스톨하고 론치하여 실행하는 역할은 MIDP에플래이어에 있는 JAM(Java Application Manager)이 이를 담당한다. <그림 4>는 JAM이 JAR와 JAD를 MIDP에플래이어에 설치하는 과정이다.

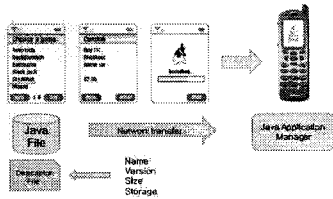


그림 4 JAM의 애플리케이션 설치 과정

MIDP의 프로그래밍에 관한 문헌으로는 강성윤, 이경범, 홍성인(2004), BizMO(2002), 오용석, 김명호, 유제정(2001), Muchow의 저서가 있으며 연구로는 김상일, 노명기, 류성열(2006)의 패스트케이스 생성기법에 관한 연구가 있다.

III. 구현 사례

3-1. 툴의 설치

미들렛 슈트를 DOS환경에서 개발하기 위해서는 JDK1.6.x와 MIDP 2.0이 필요하며 이들을 입수하기 위한 사이트와 파일은 <표 1>과 같다.

표 1. 설치 툴과 다운로드사이트

설치툴	사이트	파일
JDK 1.6.x	http://java.sun.com/javase/downloads/index.jsp	jdk-6u13-windows-i586-p.exe
MIDP 2.0	http://java.sun.com/javame/index.jsp	midp-2.0-src-window-s-i686.zip

JDK의 실행파일을 실행시키면 C:\Program Files\java\jdk1.6.0_13 디렉터리에 JDK가 설치된다. JDK설치 후 환경변수를 설정해 주어야 하는데 내컴퓨터 -> 속성 -> 고급 -> 환경변수 -> 시스템변수 -> 새로만들기에서 <표 2>와 같이 설정해준다.

표 2 JDK 환경변수 설정

변수이름	변수값
path	C:\Program Files\java\jdk1.6.0_13\bin
classpath	C:\Program Files\java\jdk1.6.0_13\lib

만약 상기의 변수이름이 존재하는 경우는 시스템변수의 편집을 선택하여 기존으로 있는 내용의 맨 마지막에 ;(세미콜론)을 추가한 후 상기의 변수값을 입력해 주면 된다. JDK의 설치가 완료되면 C:\j2me\midp2.0fcs 디렉터리를 만들고 <표 1>의 MIDP 2.0을 설치해준다. JDK와 마찬가지로 MIDP도 환경변수를 설정해 주어야 하는데 이의 내용은 <표 3>과 같다.

표 3 MIDP 환경변수 설정

변수이름	변수값
path	C:\j2me\midp2.0fcs\bin;
classpath	C:\j2me\midp2.0fcs\classes;
MIDP-HOME	C:\j2me\midp2.0fcs

JDK의 환경변수와 다른 점은 MIDP의 경우 변수값의 맨 마지막에 ;(세미콜론)과 .(마침표)가 추가된다는 점이며 MIDP의 적절한 실행을 위해서는 이에 많은 주의가 필요하다. JDK와 MIDP가 설치된 후 이를 확인하기 위해 DOS커맨드 창에서 <표 4>의 명령어를 입력하여 설치가 적절히 되었는지를 확인한다.

표 4. JDK와 MIDP 설치확인 명령어

설치 툴	명령어
JDK	java -version
preverify	preverify
MIDP	midp -version

C:\j2me\midp2.0fcs 에 MIDP를 설치하면 <표 5>와 같은 디렉터리가 새로 나타나는데 이들을 기능과 내용을 살펴보면 다음과 같다.

표 5. MIDP의 디렉터리들

디렉터리	설 명
\appdb	png 타입의 그래픽 파일 내장
\bin	preverify.exe와 midp.exe내장
\build	MIDP빌드를 위한 Makefiles
\classes	컴파일시사용할MIDP클래스들
\docs	MIDP안내서, 참고자료등
\examples	데모용 JAR와 JAD파일들
\src	데모용 소스코드
\tools	미들렛슈트 사용 위한 JAD툴

3-2. MIDlet의 생성

본 연구를 위한 미들렛 생성을 위해서 C:\j2me\projects\hello 디렉터리 밑에 Hello.java 파일을 만들어 저장하고 DOS 명령어 프롬프트에서 <표 6>의 미들렛 컴파일 명령어를 입력하여 Hello.java파일을 컴파일하여 Hello.class의 클래스 파일을 생성한다.

표 6. 미들렛 컴파일 명령어

```
javac target 1.6 -bootclasspath %MIDP_HOME%\classes Hello.java
```

Hello.class 의 미들렛 파일은 실제의 모바일장치나 MIDP에뮬레이터의 메모리구조에 맞는 클래스 인지를 사전 검증하는 preverify 절차를 거쳐서 Hello.class 파일을 재구축해야 하는데 클래스파일이 재구축되면 이전의 클래스 파일은 덮어쓰기 되어 버린다. <표 7>은 C:\j2me\projects\hello의 DOS 프롬프트에서 사전검정을 할 명령어이다.

표 7. preverify 명령어

```
preverify -cldc -classpath %MIDP_HOME%\classes; -d . Hello
```

사전검정명령어에 들어있는 ;(세미콜론), .(마침표), 스페이스(띄워쓰기) 등에 유의하여야 한다. 이렇게 사전 검정 작업이 완료되면 MIDP 에뮬레이터를 띄워서 미들렛 파일을 구현하면 된다. <표 8>은 DOS명령어 프롬프트에서 입력하여 실행할 명령어이다.

표 8. MIDP에뮬레이터 구동 명령어

```
midp -classpath . Hello
```

MIDP 명령어에서도 명령어의 옵션과 미들렛 파일 이름 사이에 .(마침표)이 있음을 유의하여야 한다. <그림 5> 는 상기 명령어의 실행 결과 이며 Hello MIDlet의 제목이 타이틀 바에 나타나 있다.



그림 5. Hello미들렛의 구현

3-3. MIDlet Suite 풀 패키지의 생성

MIDlet 생성과 관련된 작업을 완성하기 위해서는 MIDlet Suite의 풀 패키지를 생성해 주어야 한다. 미들렛 슈트는 두 개 이상의 미들렛들로 구성된다. 본 연구에서는 이를 위해 Hello2 라는 두 번째 미들렛을 하나 더 만들어 Hello2.java 로 C:\j2me\projects\hello 디렉터리에 저장한 후 전술한 컴파일 과정을 반복하여 Hello2.class 의 미들렛 클래스 파일을 생성하였다. Hello2.class 의 클래스 생성을 위한 컴파일 명령은 <표 9>와 같다.

표 9. Hello2 미들렛 컴파일 명령어

```
javac target 1.6 -bootclasspath %MIDP_HOME%\classes Hello2.java
```

앞의 경우와 마찬가지로 새로 생성된 Hello2.class 파일은 MIDP에뮬레이터에 적절한 클래스 파일인지의 여부를 사전 검증하여야 하는데 이를 위한 preverify 명령은 다음과 같다.

표 10. Hello2의 preverify 명령어

```
preverify -clde -classpath %MIDP_HOME%\classes;. -d . Hello2
```

preverify 가 Hello2.class 파일을 사진 검정하여 새로운 클래스 파일로 기존의 클래스 파일을 덮어쓰고 나면 <표 11>과 같이 MIDP 에뮬레이터의 구동 명령을 실행한다.

표 11. MIDP에뮬레이터 구동 명령어

```
midp -classpath . Hello2
```

앞의 컴파일 과정과 다른 점은 명령어는 같고 자바파일의 이름만 Hello 에서 Hello2 로 변경되었다. 이의 구현 결과는 <그림 6>과 같다.



그림 6. Hello2 미들렛의 구현

앞의 구현 결과와 같이 MIDP에뮬레이터의 타이틀 바에 Hello2 MIDlet의 제목이 나타난다.

3-3-1. Manifest 와 JAR의 생성

미들렛 슈트를 생성하기 위한 다음 단계로 앞서 생성한 미들렛 파일들을 가지고 JAR의 압축과 일을 만들어야 한다. 이를 위해서는 우선 manifest 파일이 생성되어 미들렛 파일들과 함께 압축되어야 한다. manifest 파일은 <그림 7>과 같으며 C:\j2me\projects\hello 디렉터리에 manifest.txt 라는 이름으로 저장한다.

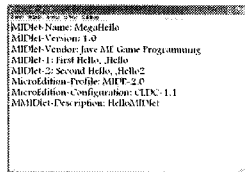


그림 7. manifest.txt 파일

JAR 파일을 생성하기 위해서는 C:\j2me\projects\hello> 의 DOS 프롬프트에서 <표

12>와 같은 명령을 입력하여 실행시킨다.

표 12. JAR 명령어

```
jar -cvfm hellosuite.jar manifest.txt *.class
```

-cvfm 인수는 현재 디렉터리에 있는 모든 클래스 파일을 manifest.txt에 근거하여 manifest와 함께 JAR 파일을 생성해 주는 옵션이다. 이 결과로 생기는 JAR파일의 이름은 hellosuite.jar파일이다. JAR 명령을 실행한 후 <표 13>의 명령을 실행시키면 JAR파일에 포함된 파일들의 내역과 압축률을 볼 수 있다.

표 13. JAR 파일의 내용 보기 명령어

```
jar -tvf hellosuite.jar
```

JAR명령어는 manifest.txt파일을 manifest.mf파일로 그 이름을 변경하였다. manifest.mf파일은 실질적인 manifest파일로 이를 생성하는데 필요한 정보를 manifest.txt파일이 제공하였다.

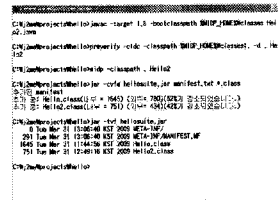


그림 8. JAR 파일의 내역과 압축률

3-3-2. JAD의 생성

미들렛 슈트의 풀 패키지를 만들기 위해서는 JAR와 manifest 파일 이외에 JAD(Java application descriptor)파일도 생성해 주어야 한다. JAD파일은 사용자가 실제로 미들렛 JAR파일을 다운로드하지 않고서도 JAR파일의 내역을 알 수 있게 해 주는 파일이다. JAD파일을 만들기 위해서는 JAR파일의 내역에다 JAR파일의 크기를 추가하여 작성한다. 본 연구에서는 노트패드로 hellosuite.jad 파일을 <그림 9>와 같이 작성하였다.

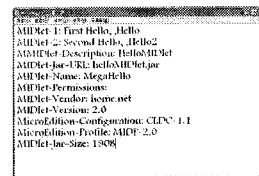


그림 9. hellosuite.jad 파일

JAD파일에는 MIDlet-jar-Size 이외에 MIDlet-jar-URL도 추가되는데 이는 JAR 파일의 위치를 가리키는 라인이다. JAD파일에서 MIDlet-1이나 MIDlet-2 는 사용자가 호출할 MIDlet을 명시하고 있으며 이들 각각은 애플레이터에서 하나의 메뉴 아이템으로 표시된다.

3-3-3. 미들렛 슈트의 폴 패키지 실행

MIDP에뮬레이터에서 미드렛 슈트를 실행시키기 위한 폴 패키지는 JAR와 JAD파일만으로 구성된다. JAR는 실제 사용자가 사용할 미들렛 클래스들과 그림, 사운드 등과 manifest.mf 파일들을 압축한 파일이며 JAD는 미들렛 JAR의 내용을 기술하고 있는 파일이다. C:\j2me\projects\hello> 의 DOS 프롬프트에서 <표 14>의 명령을 입력한다.

표 14. JAR 파일의 내용 보기 명령어

```
midp -classpath . -Xdescriptor hellosuite.jad
```

본 사례의 미들렛 슈트는 hello와 hello2의 두 개의 미들렛 클래스로 구성되어 있으며 상기 명령을 실행할 경우 JAD 파일이 다시 컴파일 되는 것이 아니라 JAD 파일이 실행될 것이라는 것만 명시한다. 애플레이터는 JAD 파일에서 명시한 파일들을 JAR파일로부터 로드시켜 실행한다. <그림 10>은 JAD 파일의 실행 결과를 보여준다.



그림 10. 미들렛 슈트의 구현

이제 MIDP 에뮬레이터의 타이틀 바에는 Select one to launch 라는 제목이 뜨고 그 아래에는 First Hello와 Second Hello가 나타나 있다 이들 각각의 미들렛을 선택하면 <그림 5>와 <그림 6>과 같은 화면이 출력된다. 즉 JAD 파일을 MIDP 에뮬레이터에서 실행시키면 초기화면으로 미들렛의 리스트들인 JAM리스트들을 보여준다는 것을 알 수 있다, JAM리스트에 클래스들을 중복해서

추가하기 위해서는 JAD파일을 <그림 11>과 같이 수정하여야 한다

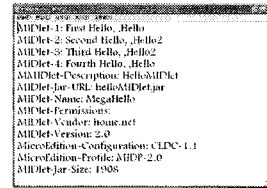


그림 11. JAD파일의 수정

상기 그림에서 MIDlet-1, MIDlet-2, MIDlet-3, MIDlet-4는 JAM리스트에 하나의 메뉴 아이템으로 각각 나타나며 hello 미들렛과 hello2 미들렛이 각각 반복되어 호출되었다.



그림 12. 수정JAD 파일의 실행 결과

JAD 파일의 변경은 hellosuite.jar를 전혀 변경하지 않고 JAD파일 내에서 호출할 MIDlet 파일만 중복 명시함으로써 수정되었다. JAD파일은 단지 JAR파일에 있는 자원을 단순히 설명하고 호출하는 역할만을 한다는 것을 알 수 있다. 실제로 서버에서 MIDP에뮬레이터로 다운로드 되어 실행되는 것은 JAD와 JAR파일이다.

IV. 연구의 의의

본 연구는 DOS 환경 하에서 J2ME프로그램의 설치와 개발과 구현의 제 과정을 사례를 통해 실행함으로써 자바 모바일프로그램의 개발과정을 순서별로 자세히 소개하였다. 이는 Java Wireless Toolkit의 프로젝트형 개발환경이나, NetBeans Mobility Pack과 같은 통합개발환경의 자동 생성 기능과는 달리 MIDlet Suite의 생성에 필요한 JAR와 JAD파일을 수동으로 명령어 하나하나씩을 실행하여 MIDlet Suite를 생성해 봄으로써 개별 개발환경이든 통합개발환경이든 융통성 있게 모바일 프로그램의 개발과 수정을 보다 쉽게 실행할 수 있는 방법을 제시하였다. 본 연구에서는 자바 모바일프로그램의 실행을 위해 JDK 1.6.x와

MIDP 2.0을 다운로드하여 설치하고 작동시켜 두 개의 미들렛 프로그램을 생성하고 이들을 Manifest파일과 함께 JAR파일로 압축하고 JAD 파일을 노트패드로 생성하여 MIDP에플레이터에서 이들을 론치하여 실행하였다. 그 결과로 에플레이터에서는 각각의 미들렛들이 JAM리스트로 나타나고 이들을 선택하면 각각의 미들렛이 실행되어 프로그램의 내용을 보여주었다. 이들의 수동적 제과정은 통합개발환경에서 자동으로 생성된 미들렛 슈트를 분해하여 수동으로 MIDP에플레이터에서 구현할 수가 있다. 다음의 연구는 이와 같은 부분을 계획해 볼 수도 있을 것이다.

참 고 문 헌

- [1] 강성윤, 이경범, 홍성인공저, “자바모바일프로그래밍: 자바 기초에서 모바일까지”, 도서출판 대립, 2004
- [2] 강덕진, “아이폰 애플리케이션 개발 프로세스와 환경구축”, 마이크로소프트웨어, 2009년4월, pp.114-119
- [3] 김동현, “모바일애플리케이션 개발 왜 흥미롭나” 마이크로소프트웨어, 2009년4월, pp.112-113
- [4] 김상일, 노명기, 류성열, “J2ME 플랫폼 기반의 패스트케이스 생성기법”, 한국정보처리학회논문지D, 제13권-D권, 제2호, 2006년 4월, pp.215-222
- [5] 김석우, “솔루션비즈니스의 침범 모바일플랫폼”, 마이크로소프트웨어, 2009년4월, pp.126-131
- [6] 김형훈, “모바일 애플리케이션을 위한 베이스캠프 모바일데이터베이스”, 마이크로소프트웨어, 2009년4월, pp.132-137
- [7] BizMO(주예찬, 김대성, 박정수, 김명훈, 최종필)저, BIPress김수, “자바모바일실전프로그래밍”, 정보문화사, 2002
- [8] 박기창, 서성채, 김병기, “J2ME MIDlet 사용자 인터페이스 자동생성을 위한 XML언어”, 한국정보처리학회논문지D, 제15-D권, 제3호, 2008년 6월, pp.327-336
- [9] 박남제, 송유진, “J2ME 기반 유무선연동의 모바일전자지갑 설계 및 구현”, 한국정보처리학회 논문지C, 제9-C권, 제5호, 2002년10월, pp.687-698
- [10] 박성서, “안드로이드 마켓이 비춰본 안드로이드 애플리케이션 개발”, 마이크로소프트웨어, 2009년4월, pp.138-143
- [11] 박영수, 장덕철, “J2ME상에서 KXML Parser를 이용한 MIDlet응용 설계 및 구현”, 한국정보처리학회논문지D, 제11-D권, 제1호, 2004년 2월, pp.211-218
- [12] 오용석, 김명호, 유제정공저, “Mobile Java Programming”, 가남사, 2001
- [13] 이창진, “아이폰 게임 개발의 실제” 마이크로소프트웨어, 2009년4월, pp.120-125
- [14] 이창진, “MIDP, 그 과거, 현재, 그리고 미래: 새로 나온 2.0과 앞으로 펼쳐질 자바 모바일의 세계”, 제4회 한국 자바 개발자 컨퍼런스, 2003
- [15] 조종근, 박윤희, 김종민, “J2ME 상에서 JSR-184를 이용한 모바일 3D 엔진의 설계 및 구현”, 한국정보과학회학술지, 제32권, 제2호, 2005년 10월, pp.673-675
- [16] 주현식, 이송원, 홍성수, WTK 개발환경 기반에서의 J2ME 3차원 모바일 응용구현” 한국컴퓨터정보학회논문지, 제13권, 제6호, 2008년11월, pp.253-261
- [17] 조종근, “모바일 3D 게임엔진을 위한 효율적인 스킨드 메시처리”, 한국게임학회논문지, 제8권, 제4호, 2008년 11월, pp.87-93
- [18] 하동욱, “m커머스로 도약하는 MIDP 2.0 분석 I”, 마이크로소프트웨어, 2003년 1월, pp.314-369
- [19] 하동욱, “m커머스로 도약하는 MIDP 2.0 분석 II”, 마이크로소프트웨어, 2003년 2월, pp.392-395
- [20] Itani, Wasim and Kayssi, Ayman I, “J2me application-layer end-to-end security for m-commerce”, Journal of Network and Computer Application, Volummn 27, Issue 1. January 2004, pp.13-32
- [21] Java Community Process, “Community Development of Java Technology Specifications: List of all JSRs”, April 2009, available at <http://jcp.org/en/home/index>
- [22] Lu, Eric Jui-Kin and Chang, “Design and implementation of a mobile database for java phones”, Computer Standards & Interface, Volumn 26, Issue 5, September 2004, pp.401-410
- [23] Muchow, John W., “Core J2ME Technology & MIDP”, Sun Microsystems, Inc., 2002
- [24] Ridley, K., “Global Mobile Phone Use To Hit Record 3.25 Billion, Reuters, June 27, 2007. available online at <http://www.reuters.com/article/email/idUSL2712199720070627>
- [25] Ritter, S. “What’s New in JavaME: MIDP 3.0, BluRay, LWUIT and more”, Sun Tech Days 2008-2009
- [26] Sun Microsystems, Inc., “CLDC 1.1 All Classes”, April, 2009, available at <http://java.sun.com/javame/reference/apis/jsr139/>
- [27] Sun Microsystems, Inc., “MID Profile All

Classes" April 15, 2009, available at <http://java.sun.com/javame/reference/apis/jsr118/index.html>

[28] Sun Microsystems, "Java SE Downloads" April 15, 2009, available at <http://java.sun.com/javase/downloads/index.jsp>