

임베디드 S/W를 위한 Java 기술 동향

조 용 윤* 김 정 훈**

◆ 목 차 ◆

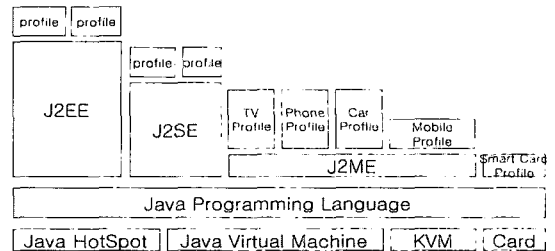
- | | |
|-----------------------------|----------------------------|
| 1. 서 론 | 3. embedded 게임을 위한 Java 기술 |
| 2. embedded s/w를 위한 Java 기술 | 4. 결 론 |

1. 서 론

1990년 제임스 고슬링(James Gosling)은 가전제품 내부에 탑재된 마이크로 프로세서를 위한 내장형(embedded) 소프트웨어를 개발하기 위한 그린 프로젝트에 투입됐다. 그러나, 기존에 사용되던 C/C++ 과 같은 범용 언어는 내장형 소프트웨어 개발에 적합하지 않아 좀더 심플하면서 빠른 반응 속도를 가지는 내장형 소프트웨어 개발용 프로그램 언어인 Java 언어를 개발하였다 [1]. 1995년에 선 소프트웨어(SunSoft)사가 Java를 발표한 이래, 인터넷과 프로그래밍 언어로서의 그 유용성이 입증된 Java는 인터넷을 기반으로 전 세계에 널리 확산되어 그 활용 범위와 위치가 점점 확고해지고 있다. 초기의 인터넷/인트라넷을 기반으로 응용 프로그램 개발에 활용되던 Java 기술은 이제 PCS, PDA, HPC(Hand-held Personal Computer), Screen Phone, GPS, 네트워크 스위치, 라우터등의 정보 통신 단말기에서 디지털 카메라, TV, VCR 등과 같은 일반 가정용 통신/전자 장비의 내장형(Embedded) 소프트웨어까지 광범위하게 쓰이고 있다. 또한, 내장형 소프트웨어를 장착한 단말기를 이용한 게임 소프트웨어에서도 Java의 활용은 활발하며, 그 범위는 광범위하다. 과거 내장형 소프트웨어의 역할은 주로 타켓 머신의 제어에 초점을 맞추고 있었다. 그러나 생활 주변의 통신/전자 기기들이 더욱 지능화 되고 네트워크화 되어야 할 필요가 생기면서

내장형 소프트웨어의 역할은 지능형 기기(Smart Device)에 서비스 개념을 불어넣기 위한 개념으로 전이되고 되고 있으며, 전자오븐, 전화기, 토스터, 공장 자동화, 운송 시스템을 비롯한 현대인의 삶에 기반을 이루는 수많은 곳에서 사용될 것으로 예측되고 있다. Java를 이용한 내장형 소프트웨어는 Java 프로그램을 내장형 디바이스(device)에 탑재함으로써 사용 가능한데, 이러한 동향은 내장형 소프트웨어 개발에서 발생하는 전통적인 문제(어려운 소프트웨어, 저가, 소프트웨어의 재사용, 실시간 지원, 소프트웨어의 업그레이드, 시장 선점을 위한 짧은 개발 기간 등)를 Java의 장점을 활용하여 해결하고 인터넷 기술을 채용한 전자제품(internet appliance)을 위한 내장형 소프트웨어를 손쉽게 개발하기 위한 것이다. 그림 1은 Java 기술의 분류와 각 Java 기술을 이용해 개발 가능한 서비스 영역에 대한 설명이다.

본 논고에서는 Java 역사와 특징에 대해 간단히 알아보고, Embedded s/w를 위해 사용될 수 있는 Java 기술에 대해 알아본다. 또한, Embedded 소프트웨어 개발



(그림 1) Java 기술의 분류와 서비스 영역(1)

* 숭실대학교 전자계산학과 박사과정

** 동서대학교 컴퓨터정보과 전임강사

을 위해 Sun이 제공하고 있는 EmbeddedJava/PersonalJava 기술에 대해 논하고, Embedded 게임 프로그램에서의 Java 기술에 대해서 논한다. 마지막으로 결론과 향후 Embedded 기술의 방향에 대해 논한다.

2. Embedded Software를 위한 Java 기술

2.1 Java의 역사와 특징

Java 언어는 1990년에 제임스 고슬링(James Gosling)이 주도한 Sunsoft 사의 Green Project에서 C++을 이용해 가전기기를 조작할 수 있는 대화형 제어기를 개발하던 과정에 만들어 졌다. C++로 개발되는 프로그램은 적용되는 제품의 하드웨어적 특성에 독립적이지 못하기 때문에 노동력과 비용이 증가하고, 또한 C++에서 제공되는 다중 상속(multiple inheritance)의 복잡성, 메모리 할당 등의 이유로 소프트웨어의 안전성, 이전 버전과의 호환성, 가격, 표준화, 효율성 등의 부분에서 많은 문제점이 있음을 인식하게 되었다. 따라서, 소프트웨어 개발자가 오직 하나의 프로그램으로 현재의 모든 플랫폼에서 실행 가능하고, 앞으로 개발될 시스템에서도 실행 할 수 있으면서 좀더 간편한 프로그래밍 언어를 개발하기로 결정했다. 초기의 Java 언어는 C++를 기반으로 하면서 C++가 갖는 복잡성과 문제점을 제거한 새로운 객체지향 언어인 Oak로 명명되었으며, 이후, Java로 불리게되었다. 1993년에는 Oak를 이용해 PDA 타입의 가전기기 제어기를 개발하였으나, 가전 시장에서의 활용범위가 적어서 기대만큼의 효과가 없었다. 1994년에 Oak 언어를 인터넷 언어로 만들려는 연구가 시도되어, 최소 자원을 이용하고 확장성이 뛰어나며, 이기종간 이식성, 호환성이 뛰어나 하드웨어나 소프트웨어 플랫폼에 독립적으로 실행이 가능한 언어인 Java가 개발되었다. 마침내 1995년 Sun은 Java와 HotJava를 공식적으로 발표했다. 새로운 인터넷 언어로써의 Java 프로그램은 특성상 컴파일 된 후, 기계 코드 대신 실행 가능한 바이트 코드를 생성하기 때문에, 프로그램이 수행되어지는 플랫폼에 독립적으로 수정 없이 수행되어 질 수 있다. 또한, Java언어는 객체 지향 언어이기 때문에, 독립적인 모듈 구현이

가능하고 코드 재사용 기회를 가질 수 있고, Interface를 지원하여 다중 상속의 효과를 가질 수 있다. 그밖에, Java 언어로 작성된 프로그램은 컴파일을 통해 바이트 코드(bytecode) 형태의 실행 파일이 되고, Java인터프리터에 의해 해석(interpret)되고 실행된다. 즉, Java언어는 컴파일 언어이자 인터프리트 언어이다. 따라서, 순수한 컴파일 언어보다는 범용성을 갖고, 순수 인터프리트 언어보다는 빠르고 효율적으로 실행된다. 마지막으로, Java 언어의 가장 큰 특징 중 하나는 뛰어난 확장성이라고 할 수 있다. Java 언어는 기존 환경과 이식성과 호환성 측면에서 이전의 범용 언어와 차별화를 가지며, 별다른 노력 없이 새로운 기능을 추가적으로 확장할 수 있다. 즉, 기존의 Java 프로그램에 새로운 기능을 확장시키기 위해서는 새로운 기능만을 코딩하거나 네트워크를 통해 해당 기능의 소프트웨어를 가져와서 기존 Java 프로그램에 플러그인(plugin)해서 사용하면 된다. 이것은 Java에서 지원하는 프로토콜 핸들러와 콘텐츠(content) 핸들러라는 개념을 이용하여 가능하게 된다.

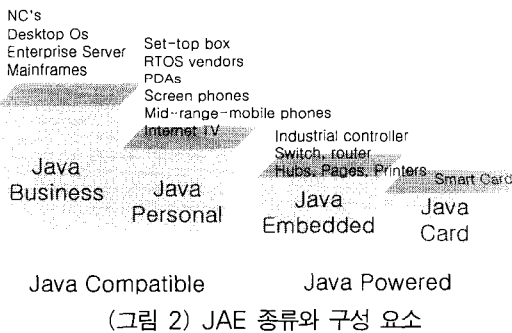
2.2 Embedded Software 와 Java

간단하게 말해, Embedded 소프트웨어는 미리 정해진 수행 기능을 빠르고, 적은 메모리 자원을 사용하며 실행할 수 있도록 하드웨어의 마이크로 프로세서에 장착한 소프트웨어를 말한다. embedded software는 여러 측면에서 필요로 되는 요구 사항을 만족해야 한다. 즉, 개발 소프트웨어의 성능에 결정적인 결함이 없어야 하기 때문에, 신뢰성(reliability)을 보장해야 하며, 보다 빠른 소프트웨어의 개발 기간을 위해 소프트웨어 재사용(software reuse)성을 보장해야 한다. 또한, 저가(low cost)의 비용으로 기존의 통신/전자 장비에 Java 내장형 소프트웨어를 탑재할 수 있어야 하며, 작은 메모리만을 이용하는 내장형 장치들의 특성을 충분히 고려해 적은 메모리 자원을 이용할 수 있도록 해야 한다. 마지막으로, 실시간 지원과 빠른 소프트웨어 업그레이드를 보장할 수 있어야 한다. Java는 대화형 가전 제품의 Embedded 소프트웨어를 개발하기 위해 개발되었기 때문에 앞서 기술한 Embedded 소프트웨어의 요구 사항을 만족시킬 수 있다. 즉, Java Embedded 소

프로그래머는 바이트 코드를 이용하는 가상 머신을 기반으로 동작하기 때문에 이식성이 높고, Java 언어의 객체 지향 특성 상, 코드 재사용과 보안성을 얻을 수 있다. 마지막으로, 가상머신이 제공하는 메모리 관리 기능인 가비지 컬렉션(Garbage Collection)에 의해 신뢰성(reliability)과 견고성(robust)을 얻을 수 있다.

2.3 JAE(Java Application Environment)

자바 기술은 크게 J2SE(Java 2 Standard Edition), J2EE(Java Enterprise Edition), J2ME(Java 2 Micro Edition)으로 나뉘며, 이때, 제공되는 JAE(Java Application Environment)는 사용 범위에 따라 크게, EnterpriseJava, PersonalJava, EmbeddedJava, CardJava로 나뉜다. 그림 2는 JAE의 구성 요소이다.



(그림 2) JAE 종류와 구성 요소

2.4 EmbeddedJava와 PersonalJava의 구성

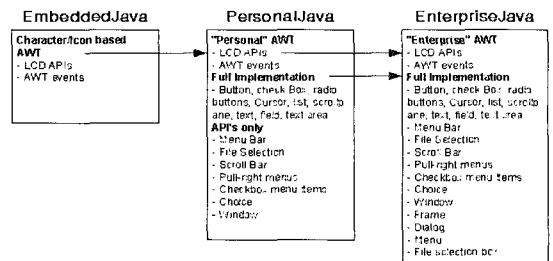
개발자가 Embedded 시스템에서 Java 기술을 사용할 수 있도록 하기 위해, Sun은 가전 제품과 소규모 통신/전자 제품을 위한 Java 플랫폼인 EmbeddedJava와 PersonalJava를 제공하고 있다. EmbeddedJava는 주로 기존의 내장형 시스템을 위한 소프트웨어에 적합하게 Java 수행 환경을 갖춘 것이다. 어떠한 의미에서는 EnterpriseJava를 축소하고 네트워크 기능을 가지지 않는 전자 제품을 위한 API를 추가한 JAE라고도 할 수 있다. 따라서, 스위치, 허브, 산업용 제어기와 같은 산업 제품에 사용될 수 있다. 반면, PersonalJava는 인터넷 환경 및 이동형 통신 기기의 범용화로 인하여 네트워크 기능을 필요로 하는 전자제품을 위한 JAE이다. Per-

sonalJava는 크기나 기능면에서 EmbeddedJava와 EnterpriseJava의 중간쯤에 위치한다고 할 수 있다. PersonalJava는 Java API로서 개인용 내장형 장비(personal consumer devices)에서 수행되는 네트워크 응용프로그램을 위한 자바 애플리케이션 환경(Java Application Environment, JAE)이다. PersonalJava는 데스크 탑 컴퓨터보다는 셋 탑 박스와 스크린폰과 같은 플랫폼을 목적으로 하고 있고, 완전한 엔터프라이즈 서비스를 필요로 하지 않는다. 따라서, PersonalJava를 기반으로 하는 제품은 set-top box, internet TV, 터치 스크린과 같은 가전 제품(consumer electronics)에서 사용된다. 또한, 적은 메모리로 동작이 가능하고, 2Mbytes ROM과 실행 시 필요할 1~2Mbytes RAM에서 동작될 수 있도록 설계되었다.

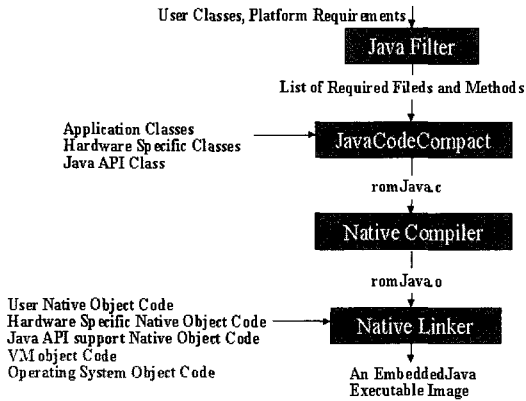
2.5 EmbeddedJava/PersonalJava의 AWT

가장 두드러진 PersonalJava와 EmbeddedJava의 차이점 중에 하나는 AWT이다. EmbeddedJava AWT는 다른 Java Application Environment들에 비해 약한 사용자 인터페이스를 제공하게 되는데, 문자 혹은 아이콘 기반의 AWT라 하여 Character-Based AWT라 부른다. 이는 EmbeddedJava의 목적이 내장형 시스템의 제어이기 때문이다. 그림 3은 각 JAE의 AWT를 비교해 놓은 것이다. PersonalJava는 EmbeddedJava와는 달리 Java 애플릿을 가지고 있어 웹 브라우징 폰과 같이 제약된 하드웨어 환경이지만 애플릿에 대한 호환성이 요구되는 환경에 사용되는 JAE임을 알 수 있다.

이전에 언급한대로 EmbeddedJava의 목표 중 하나는 메모리 크기를 줄이는데 있다. 이것을 위해 Sun은 JavaFilter와 JavaCodeCompact를 제공한다. JavaFilter는 개발자가 개발한 소스코드를 분석하여 사용되는 클래



(그림 3) 각 JAE들간의 AWT 비교



(그림 4) EmbeddedJava를 이용한 내장형 시스템 생성(2)

스, 필드, 메소드를 알아낸다. 다음에 JavaCodeCompact가 분석된 정보를 통해 사용하지 않는 모든 소프트웨어 구성요소들을 제거한 ROM 이미지를 만들어 내고, 이것을 타겟(target) 내장형 시스템의 마이크로 프로세서에 장착하면 내장형 시스템이 완성된다. 그림 4는 EmbeddedJava를 이용해 내장형 시스템을 완성하는 과정을 나타낸다.

지금까지 내장형 소프트웨어를 위한 자바 기반의 기술들에 대해 알아보았다. 다음 장은 내장형 소프트웨어중 게임 소프트웨어를 위해 사용할 수 있는 Java 기술에 대해 알아보도록 한다.

3. Embedded 게임을 위한 Java 기술

무선인터넷을 이용한 Embedded 게임의 종류를 나누는 방법은 다양 하지만, 그림 5에서 보인 것과 같이 나눌 수 있다.

초기의 Embedded 무선인터넷 게임은 단말기가 제조될 때부터 미리 게임이 삽입되어 출시된 게임을 말했다. 이러한 유형은 사용자 선택의 폭이 제한적이고 어떠한 트래픽도 발생시키지 않는다는 특징을 갖고 있기 때문에 서비스의 개념으로 바라보기는 어려웠다. 하지만 최근, 무선인터넷 게임 개발사들이 embedded 게임에서도 트래픽을 발생시키고 있다. InFusio는 2개의 embedded 게임을 프랑스 텔레콤 모바일과 미쓰비시에게 제공하고 있는데, 이러한 게임들은 SMS(Short

	플랫폼	기반언어	데이터 표현 방식	요소 기술	이동통신 사업자
1세대	WAP	WML	Browser	Gateway Browser	011/017/019
	ME	M-HTML	Browser	Browser	016/018
2세대	J2ME	JAVA	Application	KVM, CLDC, MIDP	019
	GVM	C 언어	Application	GVM	011/017
	XVM	JAVA	Application	XVM, SKT Service API	011/017
	MAP	C 언어	Application	MAP SDK	016/018
	BREW	C 언어	Application	AEE	016/018

(그림 5) 모바일 게임 플랫폼(3)

Message Service)를 통해 트래픽을 발생시키고 있다. 또한, 이러한 embedded 게임은 이 기종간의 호환성과 속도의 향상을 위해 Java 기반으로 개발될 수 있다. Java 기반 게임은 Virtual Machine 이 단말기에 내장되어 있어, 최소한의 네트워크 이용으로 성능이 메모리 용량이 낮고 성능이 데스크탑에 비해 떨어지는 모바일 기기에 사용될 수 있다. 이와 같이 자바 기반 게임은 게임 프로그램을 다운받은 후 단말기에서 단독으로 사용하거나, 네트워크에 접속하여 즐기는 게임이다. 따라서, 단말기만 많이 배포된다면 자바 기반의 네트워크 게임이 무선인터넷 게임의 주류를 이룰 수 있을 것이다. 또한, 자바 기반의 게임은 단말기에서 독립적으로 개발이 가능한 특징을 가지고 있으며, 일반 사용자들은 손쉽게 테트리스와 같은 간단한 게임들의 소스를 인터넷상에서 찾아볼 수 있고, 이를 이용하여 개인이 게임을 수정 및 보완이 가능하다. 특히, 자바를 채용한 단말기에서는 콘텐츠의 표현력이 높아 추후에 독자적인 어플리케이션을 추가할 수 있다. 자바는 다운로드형 게임과 애니메이션 등 콘텐츠의 표현력을 높일 뿐만 아니라 시큐리티 강화에도 사용된다. 예를 들면, 사용자의 단말마다 개별 정보를 부가한 자바 애플릿을 제공하여 사용자 인증에 사용하면 인증 수준이 향상될 것으로 기대된다. 그림 5는 모바일 게임을 실행하기 위한 플랫폼에 대한 소개이다. Java 기반의 embedded 게임을 위해서 EmbeddedJava를 이용하며, 이러한 embedded 게임은 통신속도와 처리성능이 향상된 IMT-2000 단말기에서 큰 역할을 할 것으로 기대되고 있다.

4. 결 론

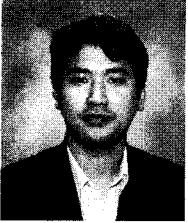
본 논고에서 인터넷 시장의 새로운 개발 플랫폼으로 확고한 자리 매김을 하고 있는 Java의 역사와 특징에 대해 살펴보고, 모바일 시장에서의 Embedded S/W에 사용되는 핵심적인 Java API를 비교 설명하였다. 또한, Embedded 게임을 위한 자바의 위치에 대해 알아보았다. Java의 소형 통신/전자 장비를 위해 EmbeddedJava/PersonalJava를 이용하려는 시도는 여러 기업과 연구소에서 활발히 진행되고 있다. 미쓰비시와 UCSC(University of California, Santa Cruz)는 발표되기 전인 EmbeddedJava/PersonalJava를 적용해 내장형 시스템을 개발하였다. 또한, 국내외적으로 메이저 통신/전자 회사인 IBM, SGI, Toshiba, Oracle, Boland, Adobe, Macromedia, Spyglass 등이 자사 제품에 Java 기술을 적용하기 위하여 Java 라이선스를 맺었고, Acorn, Chorus, GeoWorks, Lucent Technologies, Microtec, Microware[7], QNX, WindRiver[5] 등과 같은 실시간 운영체제 업체들이 Sun의 EmbeddedJava와 PersonalJava를 라이선스 함으로써, EmbeddedJava와 PersonalJava를 이용한 모바일 시장에 등장하고 있다. 실시간 운영체제 시장의 상당 부분 이상을 차지하고

있는 EmbeddedJava와PersonalJava 라이선스 업체들의 활발한 도약에 힘입어, 내장형 시스템 시장에서의 Java의 역할은 더욱 커질 것으로 보인다. 또한, Embedded 게임 업체에서의 보다 빠르고 안정적인 실행을 위해 J2ME 기술을 이용하려는 움직임은 두드러지게 나타나고 있으며, 모바일 장비를 이용한 Java 플랫폼 기반 게임 어플리케이션 서비스가 전체 무선 인터넷 게임 시장에 활기를 불어넣고 있기 때문에, 모바일 Embedded 게임 개발에 대한 Java 기술 적용도 활발히 진행될 것으로 예상된다.

참고문헌

- [1] <http://www.openmobilealliance.org>
- [2] <http://www.knet.or.kr/knet98/data/a/A23/sld012.htm>
- [3] <http://www.javacommunity.org/>
- [4] http://caic.cnu.ac.kr/youngsik/init/info_doc/tech_java.html
- [5] <http://www.wrs.com/html/trjava.html>
- [6] JavaOne : Sun's 1997 World Wide Java Developer Conference, April, 1997.
- [7] <http://www.cse.ucsc.edu/research/embedded/index-new.html>

◎ 저 자 소개 ◎



조 용 운

1995년 시립 인천대학교 전자계산학과 졸업(학사)
1998년 숭실대학교 대학원 전자계산학과 졸업(석사)
1999년~현재 : 숭실대학교 전자계산학과 박사과정
관심분야 : 프로그래밍 언어, 컴파일러, XML, HCI



김 정 훈

1996년 수원대학교 전자계산학과 이학사
1998년 숭실대학교 컴퓨터학과 공학석사
2000년 고려대학교 컴퓨터학과 박사과정 수료
1988년~2001년 8월 고려대학교 컴퓨터과학연구소 연구원
2000년 3월~2001년 8월 용인송담대학 컴퓨터소프트웨어과 겸임교수
2001년 9월~현재 : 동서울대학 컴퓨터정보과 전임강사
관심분야 : 분산/병렬처리, 클러스터링 컴퓨터, 망관리