

이동통신 무선인터넷 플랫폼의 어플리케이션 상호 호환을 위한 변환 솔루션 개발

Development of Conversion Solutions for Interoperability of Applications on Different Mobile Internet Platforms

강경보*, 강동현*, 홍창표*, 류종민**, 이종훈***, 윤정환***, 좌정우*
제주대학교 통신컴퓨터공학부*, (주)Daum Communications**, (주)WISEGRAM***

Kyung-Bo Kang(namiha@cheju.ac.kr)*, Dong-Hyun Kang(hiro4911@cheju.ac.kr)*,
Chang-Pyo Hong(cyem8@cheju.ac.kr)*, Jong-Min Ryu(dicman@daumcorp.com)**,
Joong-Hoon Lee(sean@wisegram.com)***, Jung-Han Yoon(jhyoon@wisegram.com)***,
Jeong-Woo Jwa(lcr02@cheju.ac.kr)*

요약

이동통신사업자는 기존의 음성서비스에서 고속 무선인터넷 서비스를 기반으로 하는 신규 사업모델 개발을 위해 고속 무선인터넷 망을 구축하고 다기능 휴대폰을 개발하여 보급하고 있다. 국내 이동통신사업자는 무선인터넷 서비스의 활성화를 위해 cdma2000-1x, EvDo에서 HSDPA로 고속 무선인터넷망을 구축하고 MP3, MPEG, 3D 게임엔진, DMB 등 이동멀티미디어 솔루션과 팝업 창과 같이 새로운 UI를 갖는 다기능 휴대폰을 개발하여 보급하고 있다. 또한, 무선인터넷 서비스의 콘텐츠 개발을 위해 WAP, ME 등의 모바일 브라우저와 WIPI, J2ME, BREW 등의 무선인터넷 플랫폼을 사용하고 있다. 그러나 이동통신사업자별로 다른 무선인터넷 플랫폼을 사용하고 있어 콘텐츠 사업자가 무선인터넷 서비스를 사용자에게 제공하기 위해서는 서로 다른 무선인터넷 플랫폼으로 어플리케이션을 개발하여야 하는 문제가 있다. 본 논문에서는 무선인터넷 플랫폼에서 개발된 어플리케이션을 다른 무선인터넷 플랫폼에서 동작할 수 있도록 하는 어플리케이션 상호 호환을 위한 변환 솔루션을 제안한다. 제안하는 어플리케이션 상호 호환을 위한 변환 솔루션은 대표적인 무선인터넷 플랫폼인 WIPI와 BREW 간 어플리케이션 변환을 위해 단일패스(one-pass) 컴파일러 방식을 사용하여 개발되었다. 개발된 변환 솔루션의 성능은 무선인터넷 핵심 서비스인 게임 어플리케이션에 적용하여 API 변환율, 프로그램 성능, 변환 프로그램 크기, 변환 소요시간에 대해 확인하였다.

■ 중심어 : | 무선인터넷 플랫폼 | 어플리케이션 상호 호환 | WIPI | BREW | J2ME |

Abstract

Cellular operators develop high speed mobile internet and multi-function cellular phones to activate new business model based on mobile internet services. Domestic cellular operators evolve their mobile networks from cdma2000-1x and EvDo to HSDPA to activate high speed mobile internet services. They also develop mobile internet platforms such as WIPI, BREW, and J2ME on multi-function cellular phones having multimedia solutions such as camera, MP3, MPEG, 3D game engine, DMB, PAN such as bluetooth, IrDA, W-LAN, and location information using GPS. But, content providers have problems of redevelopment of the same mobile internet application on different mobile internet platforms provided by cellular operators. In this paper, we develop conversion solutions for interoperability of mobile internet applications on WIPI and BREW using an one-pass compiler. We confirm the performance the proposed conversion solutions for the API conversion rate, the converted file size, and the full conversion time using the popular mobile games which are the killer applications on WIPI and BREW.

■ keyword : | Mobile Internet Platform | Interoperability of Applications | WIPI | BREW | J2ME |

I. 서론

이동통신망이 고도화되고 다기능 휴대폰이 보급됨에 따라 이동통신망 서비스는 기존의 음성 서비스에서 모바일 브라우저와 이동멀티미디어 플랫폼을 기반으로 하는 신규 사업모델이 개발되고 있다. 최근 이동통신사업자는 방송, 금융, 텔레매틱스 등 산업간 융합을 기반으로 하는 신규 사업 모델을 적극적으로 개발하고 있다. 또한 다기능 휴대폰의 보급과 USN, PAN 등 신규 통신망이 도입되면서 시공간 제약이 받지 않는 유비쿼터스 환경에서 서비스가 개발되고 있다[1][2].

무선인터넷 사업의 가치사슬은 단말 제조사와 더불어 고객에 지불하는 이동통신사용료를 배분하는 이동통신사업자, 인터넷 솔루션 사업자, 콘텐츠 사업자로 구성된다. 콘텐츠 사업자는 이동통신사업자가 제공하는 모바일 플랫폼을 사용하여 다양한 모바일 어플리케이션을 고객에 제공하여 수익을 창출한다. 국내 콘텐츠 사업자는 SKT, KTF, LGT의 무선인터넷 플랫폼을 사용하여 모바일 어플리케이션을 개발한다. 이 과정에서 하나의 모바일 어플리케이션을 사업자별로 개발해야 하는 부담을 갖게 된다. 국내에서는 이와 같은 콘텐츠 사업자의 어려움을 덜어 주기 위해 표준 무선인터넷 플랫폼으로 WIPI(Wireless Internet Platform for Interoperability)를 개발하여 현재 버전 2.0의 WIPI 플랫폼이 보급되고 있다. 하지만 국내에서는 WIPI 플랫폼 이외에 J2ME, BREW(Binary Runtime Environment for Wireless) 플랫폼을 병행하여 사용하며, 국내에서 개발된 모바일 어플리케이션을 수출하거나 해외에서 개발된 어플리케이션을 도입할 때 이동통신 사업자의 무선인터넷 플랫폼에 맞추어 새로이 개발해야 하기 때문에 앞의 문제가 여전히 상존한다. 모바일 어플리케이션 수출을 위해서는 WIPI 어플리케이션을 Qualcomm의 BREW, NOKIA의 Symbian, JAVA의 J2ME 등으로 변환해야 할 것이며 역으로 모바일 어플리케이션의 수입을 위해서는 BREW, Symbian, J2ME 어플리케이션을 WIPI 플랫폼의 어플리케이션으로 변환해야 한다.

본 논문에서는 WIPI 어플리케이션을 BREW 플랫폼으로, 역으로 BREW 어플리케이션을 WIPI 플랫폼으로

변환하는 솔루션을 제안한다. 2장에서는 본 논문에서 사용되는 WIPI와 BREW 플랫폼의 특징과 모바일 어플리케이션 변환 방법에 대해 설명한다. 3장에서는 WIPI Jlet 어플리케이션에서 BREW 플랫폼으로의 변환을 중심으로 이기종 무선인터넷 플랫폼 간 어플리케이션 상호 호환을 위한 변환 솔루션에 대해 기술하고 4장에서 결론을 내린다.

II. 관련 연구

1. 연구동향

[표 1]은 Java와 C 로 개발된 일반 컴퓨터의 어플리케이션 상호 변환 솔루션을 나타낸 것이다. Java->C 변환 솔루션은 Dinkumware, Ltd & EDG (Edison Design Group) Connection에서 연합 프로젝트로 개발한 것[3]과 에리조나 대학의 Sumatra프로젝트로 시작하여 개발된 Toba[4]가 있다. C -> Java 변환은 1996년에 개발된 C2J Converter 솔루션[5]이 있다.

표 1. 기존 어플리케이션 변환 솔루션

변환 대상	솔루션 이름	비고
Java -> C	Dinkumware & EDG	Java의 중간언어를 C 코드로 변환 Core Library (Java VM 기능)
	Toba	Java의 소스를 C 소스로 변환
C -> Java	C2J Converter	파일 단위로 C 소스를 Java 클래스로 변환 표준 ANSI-C, K&R C 지원

Dinkumware, Ltd & EDG Connection의 연합 프로젝트에서 개발한 솔루션은 Java의 중간언어인 바이트 코드를 C 코드로 변환하는 특징을 가지고 있다. 이때 변환된 코드는 Core Library를 사용하게 되며, 이 라이브러리는 Java VM의 Library와 같은 기능을 수행하여 Java API를 사용한 것과 동일한 기능을 수행할 수 있도록 한다.

Toba는 Java의 소스 파일을 C의 소스와 헤더 파일로 변환하고 나머지는 Dinkumware, Ltd & EDG Connection의 연합 프로젝트와 동일한 과정으로 변환하여 소스를 작동하게 한다. 그러나 일부 구현되지 않

은 기능에 대해서는 Java VM을 사용하도록 하여 Java VM이 실행이 요구될 수 있다.

C2J Converter 솔루션은 C의 소스를 파일단위로 Java의 클래스로 변환하며 ANSI-C와 K&R C 두 표준 모두를 지원한다.

무선인터넷 어플리케이션은 이동통신사업자가 제공하는 다양한 플랫폼에 맞춰 개발해야 하는 어려움이 있다. 본 논문에서 제안하는 솔루션은 위의 변환 솔루션의 기술에 착안하여 WIPI, BREW 플랫폼에 대해 개발하게 되었고, 두 플랫폼 간의 양방향 변환을 위해 동일한 언어 간 변환과 이질언어 간 변환 두 기능을 통합하였다.

2. 무선인터넷 플랫폼

본 논문에서는 국내 이동통신사업자가 사용하고 있는 무선인터넷 플랫폼인 WIPI와 BREW를 대상으로 상호 변환 솔루션을 개발하였다. [표 2]는 WIPI와 BREW 플랫폼 간의 차이점을 나타낸 것이다.

표 2. WIPI와 BREW 플랫폼 간 차이점

구 분	BREW	WIPI Clet	WIPI Jlet
사용 가능 언어	C++, C	C	Java
Device 지원	Interface	Handle	Class
메모리 할당	직접 메모리 할당	간접 메모리 할당	없음
메모리 관리기능	없음	Compaction	Garbage Collection
전역/static 변수	사용불가	둘 다 가능	static
Thread	사용 불가	사용 불가	지원
리소스 지원 형태	리소스 에디터 제공 BREW 전용 포맷	리소스 에디터 제공 JAR 파일	

[표 2]에서는 WIPI가 Clet과 Jlet을 지원하기 때문에 3가지 항목으로 나누어 비교하였다. 사용 가능 언어는 BREW와 WIPI Clet은 C 계열을 지원하고 WIPI Jlet은 Java를 지원한다. 메모리 관리기능에서 WIPI Jlet은 Java의 가비지컬렉션을 지원하고 WIPI Clet은 Compaction 기능을 갖는다. BREW는 별도의 메모리 관리기능을 지원하지 않는다[7][8]. WIPI Jlet은 Java와

같이 전역변수를 사용할 수 없지만 Static 변수는 지원하고, WIPI Clet은 C를 기반으로 하여 전역변수, Static 변수 모두를 지원한다[8]. BREW는 전역변수와 Static 변수를 지원하지 않는 대신 AEEApplet의 구조체를 통해서 유사기능을 제공하고 있다[9]. 마지막으로 가장 큰 차이점인 Thread에서는 WIPI Jlet만이 Thread를 지원한다. Thread는 적은 리소스로 다중 처리를 하는 모바일 단말기에서의 필요악으로 WIPI Clet과 BREW의 어플리케이션에서는 지원하지 않는다[7][10].

2.1 WIPI 플랫폼

WIPI는 ETRI와 정보통신부, 한국 무선인터넷 표준화 포럼 등이 공동 개발한 무선인터넷플랫폼 표준 규격이다. 2005년 4월부터 출시되는 휴대폰에 WIPI 플랫폼 탑재가 의무화가 되었고 WIPI 버전 2.0 휴대폰이 보급되고 있다.

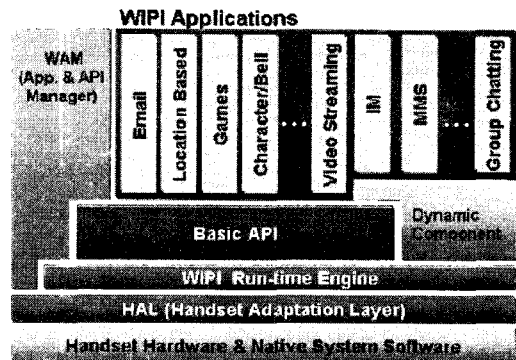


그림 1. WIPI 플랫폼 구조도

[그림 1]은 WIPI 플랫폼의 개념적 구조를 나타낸 것이다. HAL(Handset Adaptation Layer)은 WIPI 플랫폼의 하드웨어 독립성을 보장하기 위한 추상화 계층으로 HAL의 상위 계층이 하위 계층의 Native System에 대해 독립적으로 구성할 수 있도록 한다. 휴대폰의 기본 소프트웨어(Native System Software)는 플랫폼이 탑재되는 기반 소프트웨어를 말한다. 기본 API(Basic API)는 모바일 어플리케이션 개발자가 사용하는 기본 API 모음으로 Java와 C API로 구성되어 있다. 콘텐츠는 기본 API를 이용하여 개발하게 된다[10].

2.2 BREW 플랫폼

미국 Qualcomm 사가 CDMA 휴대폰에서 무선인터넷 서비스를 제공하기 위해 개발한 무선인터넷 플랫폼이다. BREW는 모바일 어플리케이션 개발을 위해 기본적으로 C++ 언어로 개발되는 API를 제공한다. 또한 BREW 플랫폼 상위 계층에 Java VM을 탑재하여 Java 언어를 사용한 모바일 어플리케이션 개발을 가능하게 할 수 있다[6].

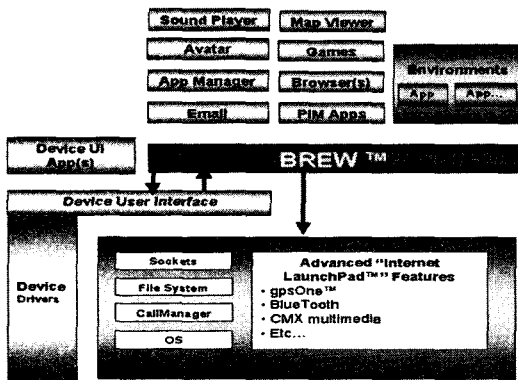


그림 2. BREW 플랫폼 구조도

[그림 2]는 BREW 플랫폼의 개념적인 구조를 나타낸 것이다. Device Driver는 휴대폰의 장치들을 사용하기 위한 소프트웨어이다. 상위 계층의 Device User Interface는 BREW 플랫폼과 하드웨어 간 독립성을 구현하며 CORE ASIC(Application-Specific Integrated Circuit) Chip Software는 별도의 Driver 없이 BREW 플랫폼이 제어 가능한 장치로 BREW 플랫폼 상에서만 지원이 된다[7][9].

3. 변환 기술

WIPI에서 BREW로 변환은 WIPI가 C와 Java API를 지원함으로써 WIPI Jlet 어플리케이션을 BREW 플랫폼으로 변환과 WIPI Clet 어플리케이션을 BREW플랫폼으로 변환 두 가지 모두를 지원해야 한다. BREW는 기본적으로 C++ API를 사용하기 때문에 WIPI Clet으로 변환만 지원하게 된다. 결론적으로 WIPI와 BREW간 모바일 어플리케이션 호환을 위한 변환 솔루션은 위의

3가지 변환을 지원해야 한다. 이 변환을 위해서는 여러 방법이 존재할 수 있다. Java VM과 유사하게 동작하는 "WIPI on BREW"와 같이 플랫폼 차원에서의 변환을 지원할 수 있지만 이는 소모되는 개발 비용과 인력에 비해 무선 인터넷 플랫폼 시장의 변화가 빨라 그 효율성이 떨어진다. 본 논문에서는 변환의 대상을 소스 코드로 하여 단말기와 플랫폼에 구애받지 않고 변환이 가능한 소스의 변환을 이용한 변환 솔루션을 제안하고자 한다.

본 논문에서 제안하는 모바일 어플리케이션 변환 솔루션은 [표 3]에서 기술된 소스 변환기술 중 컴파일 기술에 의한 Transcode 방법을 사용한다.

표 3. 소스 변환 기술 비교

변환 방안	변환결과	장점	단점
컴파일 기술에 의한 Transcode	Source Code	<ul style="list-style-type: none"> 변환 내용 검토가 용이 비교적 투명한 검증 	<ul style="list-style-type: none"> 문제 발생 시 원인 규명이 명확하지 않음
소스의 직접 변환	Source Code	<ul style="list-style-type: none"> 변환 후 디버깅 가능 원본 source의 흐름을 유지한 상태에서 변환 	<ul style="list-style-type: none"> 완벽한 1:1 mapping 불가 프로그래머 코딩 스타일에 의존적
원본 소스의 무변형 실행 (Library 사용)	Binary File	<ul style="list-style-type: none"> 변환과 관련한 디버깅의 작업의 최소화됨 	<ul style="list-style-type: none"> Library에서 오류 발생 시 해결이 어려움

소스 변환 기술은 [표 3]에 나타난 것과 같이 컴파일 기술에 의한 Transcode 방법, 소스의 직접 변환, 원본 소스의 무 변형 실행 등이 있다. 소스의 직접변환 기술은 소스를 대상 플랫폼의 소스로 변환하기 때문에 디버깅과 오류제어가 쉽다는 장점이 있다. 그러나 소스의 직접변환 기술은 정형화된 소스에 대해서는 적합하지만 개발자의 다양한 코딩 스타일을 수용하는 것은 어렵다. 즉, 개발된 소스 파서가 소스 코드의 의미를 부적합하게 변환하면 전혀 다른 소스로 변환되어 동작이 불가능 할 수 있다.

원본 소스의 무 변형 실행은 원본 소스와 Wrapper Library를 통합하여 컴파일하는 방법이다. 이 방법은 Wrapper Library가 완벽하게 기존 플랫폼과 변환 대상 플랫폼을 지원하여야 한다. 이는 Java 소스가 Java VM

이 설치된 어떤 장치에서도 동작 가능하도록 하는 것과 유사하다. 그러나 플랫폼에서 지원하는 언어가 다르면 Wrapper Library 구현이 불가능 하다. 따라서 앞에서 설명한 두 가지 변환기술은 WIPI와 BREW 간 변환 솔루션에 사용하기는 어렵다.

본 논문에서 사용하는 컴파일 기술에 의한 Transcode 방법은 소스를 대상 플랫폼의 언어로 변환하고, 이를 Wrapper Library와 통합하여 컴파일 하는 방법으로 앞의 두 가지 기술을 혼합한 것이다. 소스 변환 과정에서는 언어의 변환과 Wrapper Library가 지원하지 않는 API가 있으면 이를 직접 변환하거나 변환 할 수 없으면 오류처리 후 주석으로 표기한다.

III. 이기종 모바일 플랫폼 간 어플리케이션 변환 솔루션

본 논문에서 제안하는 WIPI와 BREW 간 상호 변환 솔루션은 WIPI Jlet->BREW, WIPI Clet->BREW, BREW->WIPI Clet 모듈로 이루어진다. 각 모듈은 소스 변환기, Wrapper Library, 이벤트 전달을 위한 Applet, 리소스 변환기로 구성된다. 본 장에서는 WIPI Jlet->BREW 변환 모듈을 중심으로 각 모듈의 구성요소에 대해 설명한다. 또한 5절의 구현 및 실험에서 각 모듈의 성능을 확인한다.

1. 시스템의 구성

[표 4]와 같이 모바일 어플리케이션 변환 모듈은 크게 4가지의 구성요소를 가진다. 어플리케이션의 소스를 대상 플랫폼이 지원하는 언어로 변환하는 변환기, 변환된 소스를 구동시켜주는 Wrapper Library와 BREW Applet, 응용프로그램에서 사용되는 리소스의 변환을 수행하는 리소스 변환기로 구성된다. 변환기는 원본 소스를 읽어 최소한의 변환을 수행한다.

대상 플랫폼에 맞춰 제작된 Wrapper Library는 원본 콘텐츠가 구동되던 플랫폼의 API를 모두 담고 있어야 하고, 그 기능과 동일하게 동작해야 하지만 플랫폼에 특성에 따라 구현이 불가하거나 불필요 시 구현의 대상에서 제외 될 수 있다. BREW Applet은 변환된 소스가

해당 플랫폼의 구동환경과 맞지 않기 때문에 어플리케이션을 구동하고 상황에 맞는 이벤트를 발생시켜 줄 최소한의 응용 프로그램이다. 이 프로그램의 핵심은 이벤트의 전달이다. 플랫폼에 맞추어 이벤트를 적절히 전달을 해줘야 변환된 어플리케이션이 정상적으로 작동할 수 있다. 전달될 이벤트도 Wrapper Library와 같이 플랫폼의 특성에 맞추어 발생하지 않는 이벤트와 상황에 따라 전달 할 필요가 없는 이벤트가 존재한다.

표 4. WIPI Jlet 어플리케이션을 BREW 어플리케이션으로 변환 시 구성요소

구성요소	구현내용
변 환 기	• Java언어를 C++언어로 변환 기능
Wrapper Library	• WIPI Jlet Class Library를 C++언어를 사용하여 구현
BREW Applet	• BREW 플랫폼에서 구동될 Applet 구현 • 모든 이벤트는 변환된 소스로 전달 기능
리소스 변환기	• 대상 플랫폼에 맞게 이미지, 사운드와 같은 리소스의 포맷을 변환한다.

2. 구동환경의 조성

[그림 3]은 WIPI Jlet 어플리케이션을 BREW 플랫폼으로 변환하여 작동될 때의 계층구조를 나타낸 것이다. BREW 플랫폼 위에 구동 환경을 맞춰줄 BREW Applet, 소스 어플리케이션이 구동되던 플랫폼의 API를 해당 플랫폼의 언어로 작성된 Wrapper Library가 위치한다. [그림 3]과 같은 구동 환경을 구성하면 변환된 소스는 기존의 플랫폼의 API를 그대로 사용할 수 있고, 이벤트의 처리 또한 기존의 플랫폼에서 작동되는 방식을 그대로 사용할 수 있다. 즉 Wrapper Library와 이벤트를 처리할 BREW Applet이 존재하기 때문에 변환기에서 원본 소스 변환이 최소화된다.

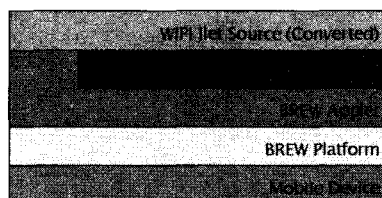


그림 3. 변환된 어플리케이션의 구동 시 계층구조

3. 소스의 변환

[그림 4]는 일반적인 컴파일의 과정을 나타낸 것이다. 본 논문에서 제안하는 변환 솔루션의 목표는 원시소스를 다른 플랫폼의 원시소스로 변환하는 것이므로 구문 트리 최적화에서 목적코드 단계까지는 필요하지 않다.

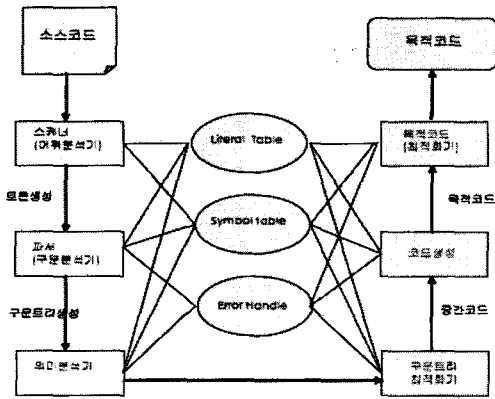


그림 4. 컴파일 과정

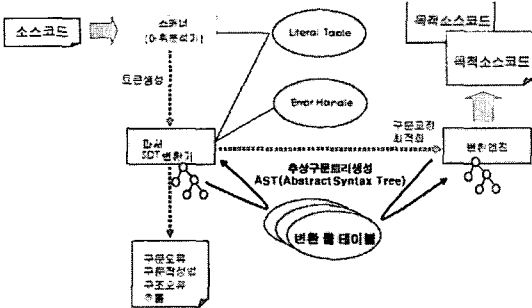


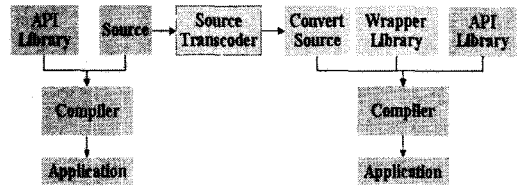
그림 5. 소스 변환의 과정

[그림 5]는 소스의 변환의 과정을 나타낸 것이다. 소스의 변환은 컴파일러 이론이 정립되기 전 초창기에 사용되었던 단일패스(one-pass) 컴파일러 방법과 동일하여 프로그램에 대한 중간코드를 생성하지 않고 직접 대상 플랫폼의 소스 코드로 변환을 수행한다. 변환기의 처리 과정을 살펴보면 소스코드는 어휘 분석기에 의해 토큰화된 원시 프로그램은 문법-지시적 변환 방법(SDT : Syntax Direction Translation)을 이용한 파서에 의해서 Bottom-up 방법으로 추상구문트리(AST)를

만들고 AST 상에 대상 플랫폼에 적합한 소스 코드로 일대일 Mapping된 정보를 사용하여 변환을 완료한다.

4. 시스템의 동작

변환 전과 후의 플랫폼에서 어플리케이션의 동작과정은 [그림 6]과 같다. 변환 플랫폼의 동작과정은 [그림 6]의 오른쪽 부분이다. 첫 번째 과정은 소스의 변환 과정으로 변환기를 통해 원본 WIPI Jlet 소스를 BREW의 C++ 코드에 맞게 변환한다. 이때 문제가 발생하면 변환기는 해당 코드의 위치와 오류메시지를 사용자에게 보여주고 변환을 계속 수행한다. 이와 같이 자동변환이 불가능한 부분은 변환과 관련된 설명과 함께 주석으로 처리된다.



< 기존 플랫폼에서 동작 >

< 대상 플랫폼에서 동작 >

그림 6. 변환 전후의 플랫폼 동작과정 비교

두 번째 과정은 변환 소스의 컴파일 과정이다. 변환된 소스는 Wrapper Library, BREW Applet과 함께 Arm 컴파일러를 통해 컴파일 된다. 이때 BREW 플랫폼의 API Library도 컴파일 과정에 포함된다. 소스 변환기의 변환과정 중 검출할 수 없는 오류 혹은 프로그래머 고유의 코딩 스타일로 인해 잘못 변환된 내용은 변환 중에서 컴파일 오류가 발생할 수 있으며, 이는 개발자가 직접 변환된 소스를 보고 수정해야 한다.

마지막 과정은 소스의 변환과는 별도로 이루어지는 작업으로 이미지, 사운드와 같은 리소스 변환이다. WIPI와 BREW에서 지원하는 이미지 형식이나 사운드 형식이 달라 해당 플랫폼에서 지원하지 않으면 그에 맞게 리소스의 포맷을 변환한다. [표 5]는 WIPI 와 BREW 플랫폼에서 지원하는 리소스의 포맷을 비교한 것이다.

표 5. 플랫폼에서 지원하는 리소스 포맷 비교

구분	WIPI 플랫폼	BREW 플랫폼
이미지 포맷	BMP, PNG, GIF	BMP, PNG, BCI
이미지 크기	크기, 색상의 제한 없음	150X150으로 제한 256색상으로 제한
사운드 포맷	MIDI, WAVE, MP3	MIDI, QCP, MP3

5. 구현 및 실험

변환기 개발은 VisualStudio 6.0의 환경에서 VC++을 이용하여 개발하였다. Wrapper Library는 WIPI Jlet API 소스를 이용하여 개발하였다. WIPI Jlet 어플리케이션에서 BREW 플랫폼으로 변환 성능을 확인하기 위해 무선인터넷 핵심서비스인 게임 중 PNJ사의 “푸시푸시 삼국지”를 사용하였다.

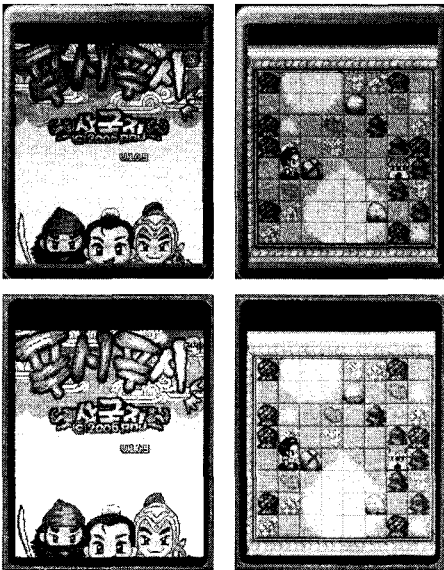


그림 7. 에뮬레이터 실행 화면 (상-WIPI, 하-BREW)

[그림 7]은 변환 전과 후의 플랫폼에서 게임 어플리케이션의 구동 화면을 나타낸 것이고 [그림 8]은 변환 솔루션의 실행 화면을 나타낸 것이다. 최종적으로 변환 솔루션에 의해 변환된 콘텐츠를 BREW 기반의 휴대폰에 다운로드 받아 게임 어플리케이션이 해당 플랫폼에서 정상 동작하는지 확인하였다.

이기종 무선인터넷 플랫폼 간 모바일 어플리케이션

변환솔루션의 성능은 [그림 9][그림 10]과 같다. [그림 9]는 Wrapper Library에 API의 구현 비율을 나타낸 것이다. [그림 10]은 Wrapper Library의 API 중 변환후의 호출 시간이 변환 전에 대해 110% 내에 값을 갖는 API의 백분율을 나타낸 것이다.

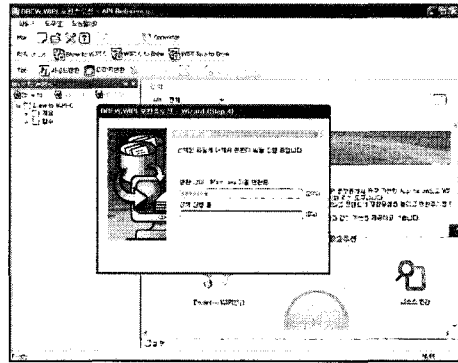


그림 8. 변환 솔루션 실행 화면

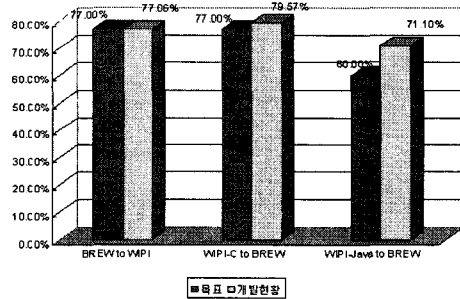
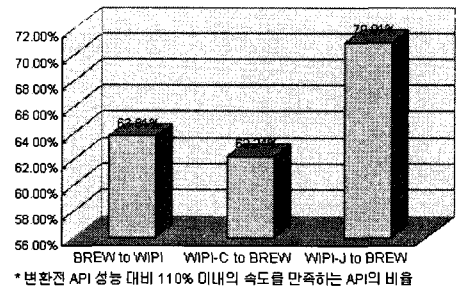


그림 9. Wrapper Library 구현율



* 변환전 API 성능 대비 110% 이내의 속도를 만족하는 API의 비율

그림 10. Wrapper Library 성능측정 결과

[표 6]은 변환 솔루션의 성능을 API 변환율, 변환된 어플리케이션의 크기, 변환에 소요된 시간 등으로 나타

낸 것이다. API 변환율은 [그림 9]를 표로 나타낸 것이고 프로그램 성능은 API 호출 시간을 호출 횟수를 고려한 평균값이다. 이 값은 API별로 어플리케이션에서 호출되는 횟수를 구하고 이에 변환 전 호출 시간에 대한 변환 후 호출 시간 비의 백분율을 곱한 값을 구한다. 구한 값을 전체 API에 대해 더한 다음 이 값을 전체 호출 횟수로 나누어 구한다. 프로그램 크기는 변환 전 어플리케이션 크기에 대한 변환 후 크기의 비를 백분율로 나타낸 것이다. 변환 소요 시간은 하나의 어플리케이션을 변환하는 데 소요되는 전체 시간을 평균값으로 나타낸 것이다.

표 6. 변환 솔루션 성능 분석표

구분	BREW to WIPI	Clet to BREW	Jlet to BREW
API 변환율	77.06%	79.57%	71.10%
프로그램 성능	71.79%	157.63%	107.98%
프로그램 크기	118.90%	123.00%	102.31%
변환 소요시간	4~5시간	6~10시간	1~4일

[표 6]의 결과를 보면 BREW->WIPI Clet, WIPI Clet->BREW, WIPI Jlet->BREW 변환 모듈에서 Wrapper Library의 구현율은 70% 이상 값을 갖는다. 또한 프로그램 성능에서 BREW->WIPI Clet은 71.79%, WIPI Jlet->BREW는 107.98%로 속도가 빠르거나 대등하다. WIPI Clet->BREW는 150% 값을 갖는데 이는 WIPI Clet API에는 BREW API에서 지원하지 않는 항목이 많아서 상대적으로 속도가 느리다. 변환된 프로그램의 크기는 전에 비해 1.3배 이상 커지지 않음을 확인할 수 있다. 변환에 소요되는 시간은 사용자의 수동변환 시간을 포함하는 것으로 개발자에 따라 더 짧아 질 수 있다. 상용 어플리케이션과 휴대폰을 이용하여 본 논문에서 제안하는 소스 코드 변환 솔루션에 대한 성능측정 결과 만족스러운 결과를 얻을 수 있었다.

IV. 결론

본 논문에서는 WIPI / BREW 플랫폼에서 개발된 모

바일 어플리케이션을 상호 변환을 위한 솔루션을 제안하였다. 제안된 솔루션은 Java 기반의 WIPI Jlet API로 이미 개발된 모바일 어플리케이션을 C++ API를 갖는 BREW 플랫폼의 어플리케이션으로 변환 모듈을 예들 들어 기술하였다. 동일한 과정을 통하여 BREW 어플리케이션을 WIPI Clet으로 변환, WIPI Clet 어플리케이션을 BREW로 변환 모듈 또한 개발하여 솔루션에 포함하였다.

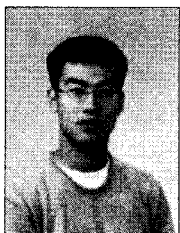
본 논문에서 제안하는 모바일 어플리케이션 변환 솔루션은 특정 플랫폼에서 개발된 어플리케이션을 다른 플랫폼에서 구동하기 위해 이중으로 개발하는 방식을 탈피하여 개발 비용과 소모되는 인력을 감소시키고, 무선인터넷 콘텐츠의 개발을 활성화하는 데 기여할 수 있다. 또한 본 논문에서 제안한 솔루션의 설계과정을 통해 WIPI, BREW 이외에 MIDP, GVM, Symbian 등 다양한 무선인터넷 플랫폼으로 확장된 변환 솔루션을 개발할 수 있을 것이다.

참고 문헌

- [1] 김성환, 양석호, *휴대폰을 위한 모바일 자바 프로그래밍*, (주)피어슨에듀케이션코리아, 2005.
- [2] 배석희, 한상홍, 전영준, *위피 WIPI 모바일 프로그래밍 기술을 통일한 위피 입문서*, 대림출판사, 2004.
- [3] http://www.edg.com/index.php?location=java_limitations
- [4] <http://www.cs.arizona.edu/sumatra/toba/>
- [5] http://tech.novosoft-us.com/product_c2j.jsp
- [6] <http://brew.qualcomm.com/brew/ko/>
- [7] 박상태, *Brew 기본 개념*, Geotel, 2001.
- [8] KTF, *WIPI 개발자 가이드*, 2004.
- [9] R. Rischpater, *Software Development for the QUALCOMM BREW Platform*, Apress, 2005.
- [10] 안창남, "WIPI SDK 기술교육 자료", AROMA SOFT, 2003.

저자 소개

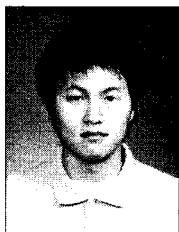
강 경 보(Kyung-Bo Kang) 준회원



- 2001년 3월 ~ 현재 : 제주대학교
통신컴퓨터공학부 (학사 과정)

<관심분야> : 모바일 프로그래밍, 이동통신 프로토콜

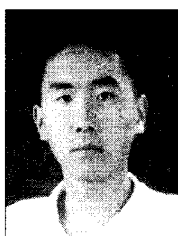
강 동 현(Dong-Hyun Kang) 준회원



- 2001년 3월 ~ 현재 : 제주대학교
통신컴퓨터공학부 (학사 과정)

<관심분야> : 모바일 프로그래밍, 3D 영상 처리

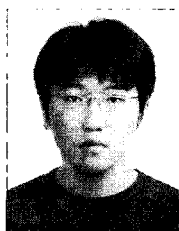
류 중 민(Jong-Min Ryu) 준회원



- 2007년 2월 : 제주대학교 컴퓨터
공학과 (공학사)
- 2006년 12월 ~ 현재 : (주)
Daum Communications 서비스
혁신본부 콘텐츠 개발팀

<관심분야> : 모바일 프로그래밍, 웹 프로그래밍

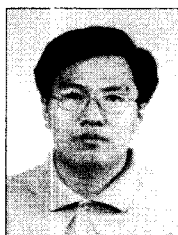
홍 창 표(Chang-Pyo Hong) 준회원



- 2007년 2월 : 제주대학교 통신공
학과 (공학사)

<관심분야> : 모바일 프로그래밍, 보안 시스템

좌 정 우(Jeong-Woo Jwa) 정회원



- 1985년 2월 : 한양대학교 전자공
학과 (공학사)
- 1987년 2월 : KAIST 전기 및 전
자공학과 (공학석사)
- 2001년 8월 : KAIST 정보 및 통
신공학과 (공학박사)
- 2002년 10월 ~ 현재 : 제주대학교 통신컴퓨터공학
부 조교수

<관심분야> : 무선인터넷 BM, 이동통신 프로토콜

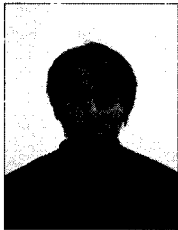
이 중 훈(Joong-Hoon Lee) 정회원



- 1995년 2월 : 한양대학교 전자계
산학과 (공학사)
- 1995년 3월 ~ 1999년 2월 :
(주)LG소프트 개발팀 (주임연구
원)
- 1999년 3월 ~ 2002년 1월 : (주)
세스컴 개발팀 (선임연구원)

- 2002년 1월 ~ 현재 : (주)WISEGRAM 기술연구소
(기술이사)
 - 2005년 1월 ~ 현재 : 한국산업기술대학교 겸임교수
- <관심분야> : 모바일 프로그래밍

윤 정 한(Jung-Han Yoon) 정회원



- 1997년 2월 : 한국외국어대학교
컴퓨터공학과 (공학사)
- 1997년 1월 ~ 2002년 2월 :
(주)웨더뉴스 개발팀 (대리)
- 2003년 10월 ~ 2005년 1월 :
(주)컨피테크 개발팀 (과장)

- 2005년 2월 ~ 현재 : (주)WISEGRAM 기술연구소
(부장)
- <관심분야> : 모바일 프로그래밍