

스마트폰 플랫폼을 위한 모바일 콘텐츠 자동 변환기

이 양 선*

1. 서 론

1.1 국내·외 현황 및 문제점과 전망

기술적으로는 최근에 프로그래머가 프로그램을 한번 작성하면 프로세서나 운영체제와 같은 플랫폼에 의존하지 않고 어느 시스템에서나 실행할 수 있는 가상기계(virtual machine)와 컴파일러 기술에 대한 연구와 가상기계를 이용하여 모바일 디바이스 상에서 동적인 애플리케이션을 실행할 수 있는 모바일 플랫폼에 관한 기술이 국내외적으로 활성화되고 있다. 또한, 무선 네트워크를 통해 단말기로 동적인 애플리케이션을 다운로드할 수 있다는 점에서 가상기계를 기반으로 모바일 플랫폼에 관한 기술에 관한 연구가 폭넓게 진행되고 있으며, 최근에 다양한 종류의 차세대 기능을 갖춘 스마트폰의 등장으로 새로운 환경을 맞고 있다.

산업적으로는 스마트폰이 확산되면서 가장 큰 산업의 변화는 패러다임의 변화이다. 기존의 음성 통화 중심에서 애플리케이션으로 전환되면서 산업구조가 통신사업자가 서비스를 주도하는 중앙

집중형에서 사업자와 무관하게 다양한 서비스 제공이 가능한 분산형으로 변화되었다. 또한, 패러다임이 전환되면서 경쟁의 원천도 개별우위 요소에서 애플리케이션 생태계 구축 역량으로 이동하고 있다. 즉, 스마트폰이 확산되면서 통신 서비스는 물론 단말기에서도 양질의 애플리케이션을 확보하는 것이 새로운 경쟁요소로 부상한 것이다. 따라서 개발자 참여가 소비자 이용을 이끌어내고 다시 더 많은 개발자가 참여하는 눈덩이 효과가 나타날 수 있도록 개발자와 소비자를 확보하여 애플리케이션 생태계를 구축하는 것이 관건이 되었다.

한편, 국내에서 일반적으로 사용하는 모바일 플랫폼에서 개발된 콘텐츠를 다른 통신사 및 해외 통신사에 서비스하기 위해서는 해당 모바일 플랫폼인 BREW, MIDP, Android, iPhone(iOS), Window Mobile 등으로의 변환작업이 필수적이며, 국내의 경우는 현재 모바일 표준 플랫폼인 WIPI나 스마트폰 플랫폼인 Android, iPhone, Window Mobile로 변환해야만 한다. 일반적으로 모바일 원 소스 콘텐츠를 분석(1개월)하여 다른 플랫폼에 재구성(1-2개월)하는데 2-3개월의 시간과 2-3명의 인건비용이 소요된다. 이러한 변환 작업을 자동화하는 솔루션을 개발함으로써 각각의 다른 플랫폼으로 전환 시 자동변환과 1~3일 정도로 보조 작업으로 변환작업을 완료하여 2-3개월의 시간과 비용을 단축 할 수 있다[25-39].

* 교신저자(Corresponding Author): 이양선, 주소: 서울시 성북구 정릉동 16-1(136-704), 전화: 02)940-7743, Fax: 02)940-7615, E-mail: yslee@skuniv.ac.kr

* 종신회원, 서경대학교 컴퓨터공학과 교수

* 이 논문은 2010년도 정부(교육과학기술부)의 재원으로 한국연구재단의 기초연구사업 지원을 받아 수행된 것임(No. 20100023644)

(1) 해외 산업현황

지난 2008년 전 세계 게임시장의 규모는 2007년 약 409억 달러에서 약 470억 달러로 14.8%의 고성장을 구가하였으며, 이 중 콘솔게임이 약 278억 달러로 전체 게임 시장의 60%, 온라인 게임이 약 80억 달러로 17.2%, 모바일 게임이 약 72억 달러로 15.4%를 차지한 것으로 추정된다. 또한, 향후 게임시장은 Apple의 iPhone 및 App Store 출시를 계기로 기존의 폐쇄적 유통구조에서 벗어나고 있는 모바일 게임시장이 가장 빠른 성장세를 보일 것으로 예상된다.

현재 전체 시장의 60%를 차지하고 있는 콘솔게임은 닌텐도 Wii와 같은 체감형 게임기의 등장으로 기존에 없던 새로운 시장이 열리고 계속 발전할 것이다. 전 세계 게임 시장은 게임업체와 이용자 그리고 관련 기술과 주변 환경의 상호 작용 속에 몇 가지 주요 트렌드를 보여주고 있는데, 이는 게임의 다양성 확대(체감형 게임의 인기, 기능성 게임의 확대, 확장된 게임성 제공 사례 증가), 수익 다각화 전략(게임 타이틀의 멀티플랫폼화, 플랫폼의 멀티서비스 증가, OSMU의 증가), 게임업체의 리스크 관리 강화(M&A 증가, 다작 전략, 맞춤형 가격 정책), 신규 시장참여자의 증가(게임 장르간 서비스 접목, New Device 게임의 출현 및 확대)의 양상으로 요약할 수 있다.

또한, 향후 전 세계 모바일 게임시장은 극심한 글로벌 경제 침체임에도 불구하고 2009년도까지 성장해 왔지만 2010년도 큰 폭으로 하락한 후에 다시 성장할 것으로 예상하고 있다. 그것은 스마트폰과 하이엔드급 휴대폰의 확대 성장에 관계된다. 그리고 Nokia나 Apple과 같은 대형 업체가 시장을 리드하며 마케팅 효과를 줄 것으로 예상되고 있다. 현재 모바일 게임 시장은 실제 고객들의 실수요보다는 마케팅 효과에 의해서 리드되고 있기 때문이다. 그림 1은 전 세계 모바일게임 시장

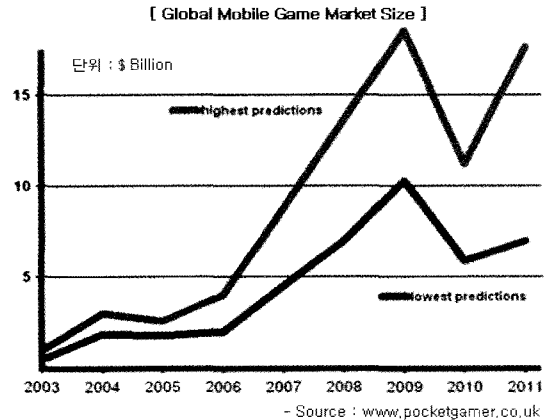


그림 1. 전 세계 모바일게임 시장 전망

전망을 나타낸 것이다.

한편, 모바일게임 유통구조가 다변화되고 있다. 즉 애플이 도입한 개방형 콘텐츠 웹 마켓인 앱 스토어(아이폰)로 대변되는 새로운 유통 경로는 모바일 게임 콘텐츠를 무선랜(와이파이)이나 PC에서 다운로드 방식을 통해 제공받을 수 있는데 애플에 이어 구글의 안드로이드 마켓(구글폰 G1)과 같은 Off Portal에서 게임 카테고리의 수요가 많아지는 것도 긍정적인 요인으로 작용하고 있다. 노키아 역시 '모시'와 '오비스토어'라는 웹 장터를 마련했으며 마이크로소프트도 '윈도 마켓 플레이스'를, 구글이 '안드로이드 마켓'을 론칭할 계획이다. 긍정적인 시나리오로서는 2009년에 스마트폰 판매량은 192.3백만대를 예상하는데 이것은 전년대비 11.1% 성장한 수치이다. 소극적인 예측으로는 183.9백만대를 예상하고 있는데, 이는 전년대비 6% 성장한 것이다. 그림 2는 세계 스마트폰 시장 전망을 나타낸 것이다[45].

(2) 국내 산업 현황

국내 모바일게임 시장은 선도 업체의 매출 규모만 크게 증가하여서 부익부 빈익빈의 심화로 인한 영세 개발사의 도산이 속출하고 있다. 이 같은 상황에서 전체의 매출이 다소 증가한 것은

ISuppli Figure: Global Smart Phone Unit Shipment Forecast (Millions of Units)

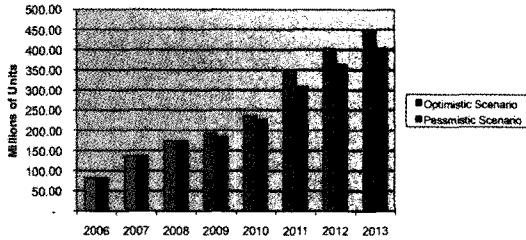


그림 2. 세계 스마트폰 시장 전망

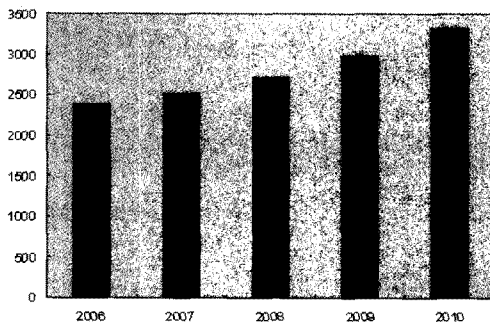
시장의 파이가 커지지 않은 상태에서 수익 환경 여건만 강화되었다고 여긴다. 뿐만 아니라, 도산한 모바일 게임 개발사들 중 대다수가 선도 기업의 하청기지로 전락했다는 사실이다. 그림 3은 국내 모바일게임 시장 규모 현황과 전망을 나타낸 것이다.

모바일게임의 경우, 상위 업체와 하위 그룹간의 양극화는 더욱 심해졌으며, 그에 따른 인수합병이 가속화되었다. 모바일 게임 유통 환경의 전반적인 구조 조정은 이루어지지 못했으며, 신규 업체의 진입장벽은 더욱 높아졌다. 모바일게임의 특성상 이동통신사의 의지가 시장을 움직이는 경향이 강하다는 점을 감안해 볼 때, 시장 불균형을 개선하기는 당분간 어려울 것이다.

그럼에도 불구하고 올해 예상되는 국내 모바일 게임 시장의 가장 큰 변화는 'WIPI 탑재 의무화 고시 폐지'이다. WIPI는 정부가 주도적으로 만든

무선인터넷 접속 플랫폼으로서 2004년부터 휴대폰에 의무적으로 탑재해야 했지만 경쟁력 약화 등을 이유로 WIPI 탑재 의무화 규정이 단계적으로 폐지되고 이후 브루, 심비안 등 다양한 무선인터넷 접속 소프트웨어를 탑재한 휴대폰들이 시장에 풀리게 된다. 문제는 국내 모바일게임사들 대부분이 지금껏 WIPI에 맞춰 게임을 개발해 왔다는 점이다. WIPI 탑재 의무화가 폐지되면 적용해야 할 플랫폼이 확대되고 포팅 비용도 그만큼 늘어나게 된다. 이미 비슷한 환경인 해외 시장 공략 경험을 가진 대형 업체들과는 달리 자금난과 인력난에 허덕이는 중소 개발사들은 그만큼 경쟁력이 약화될 수밖에 없다.

그리고 국내에서도 역시 모바일게임 유통구조가 다변화되고 있다. 개방형 콘텐츠 웹 마켓의 도입을 위해 SK텔레콤은 2009년 4월부터 스마트폰 서비스사이트인 마이스마트(www.mysmart.co.kr)와 'T앱스토어'를 통해 게임 등을 유료 판매할 예정이다. 또한, 국내 단말기 제조사인 삼성전자(삼성 모바일 이노베이터), LG전자(LG 모바일 개발자 네트워크) 등도 표면상으론 단순 소프트웨어 개발자들의 편의를 도모하기 위한 커뮤니티를 만든 것이나 새롭게 열리는 모바일 콘텐츠 유통 사업에 진입하려는 것으로 여긴다. 이 같은 변화는 그동안 이동통신사 중심의 폐쇄적인 무선인



구분	매출액(억원)	성장률
2006	2390	-
2007	2518	5.4%
2008	2719	8.0%
2009	2991	10.0%
2010	3320	11.0%

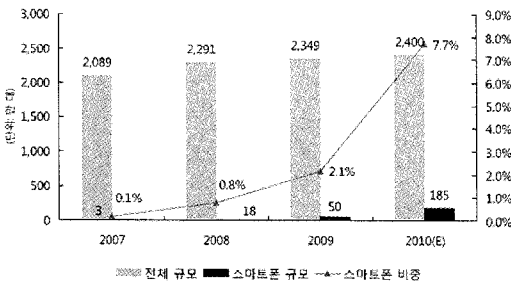
그림 3. 국내 모바일게임 시장 규모 현황과 전망

터넷 환경을 수용할 수밖에 없었던 국내 게임사들의 유통경로를 확장시켜 줄 것이다. 또한 EA 모바일, 게임 로프트 등 외국계 거대 모바일 게임 업체들의 국내 스마트폰 시장에서 주목받고 있다. 이들 업체는 이미 다수의 스마트폰용 게임 콘텐츠와 제반 유통 여건까지 확보하고 있어서 스마트폰용 게임 콘텐츠 시장 선점에 매우 유리한 고지를 점하고 있다[25-39,45].

상기 서비스 플랫폼 개발 전략으로 개인 및 기업 무선 인터넷 서비스 시나리오를 진행중이다. 또한, 퀄컴이 무선 통신 업계에서의 탄탄한 배경을 기반으로 출시한 BREW는 C/C++뿐만 아니라 HP의 MicroChaiVM을 통해 자바를 지원할 것을 표명해 기존의 무선 인터넷 플랫폼 업체들에게 막대한 영향력을 미칠 것으로 판단된다.

한편, 스마트폰은 기본적인 휴대폰 기능에 컴퓨터와 같은 기능이 포함된 차세대 핸드폰으로 시장의 각광을 받으며 널리 확산되고 있는 실정이다. 스마트폰의 플랫폼으로는 Google이 Java를 사용할 수 있는 Android 플랫폼을, Apple이 C/Objective-C 언어를 사용할 수 있는 iPhone용 iOS를, 마이크로소프트가 C/C++, C#, Java 등을 사용할 수 있는 Windows Mobile 플랫폼을, 그리고 노키아가 C/C++, Java 등을 사용할 수 있는 Symbian 플랫폼을 발표함으로써 새로운 도전에 직면하고 있다. 그러나 이러한 모든 기술이 휴대폰 단말기와 같은 분야에만 집중되어 있지 다른 IPTV, PMP, PDA 등의 임베디드 시스템과 같은 컨버전스 플랫폼으로는 확산되지 못한 실정이다.

국내 휴대전화 시장 전체 규모 및 스마트폰 비중 전망. 단위: 만대



Source: ROA Group Korea 추정치
Note: 법인용 스마트폰 포함

그림 4. 국내 휴대전화 시장 규모 및 스마트폰 비중 전망

(3) 해외의 기술동향 및 수준

국외 모바일 플랫폼으로는 일본 NTT 도코모의 i-애플리가 비표준 자바 규격인 DoJa, 자체 API 적용 등으로 모바일 플랫폼 서비스의 독자적인 영역을 확보해 나가고 있으며, 노키아는 대규모의 자바 지원 단말기를 활용하고 있는 실정이다. 마이크로소프트사는 스타팅 기반의 차세대 휴대용 단말기 및 운영체제 개발과 .NET이라는 가

(4) 국내의 기술동향 및 수준

지금까지 국내에서 발표된 모바일 플랫폼과 스마트폰 플랫폼은 표 2, 표 3과 그림 5와 같이 구분할 수 있다. 국내 이동 통신사들의 활발한 모바일 플랫폼의 도입은 다양한 어플리케이션과 솔루션 등장을 촉진시켜 전 세계 모바일 시장을 선점할

표 1. 해외 스마트폰 플랫폼 및 개발언어

	플랫폼	개발언어	마켓플레이스
마이크로소프트	Windows Mobile 6.5	C/C++, C#, Java	Windows Marketplace for Mobile(SkyMarket)
애플	iOS(iPhone)	Objective-C	AppStore
구글	Android	Java	Android Market
노키아	Symbian	C/C++, Java	Ovi

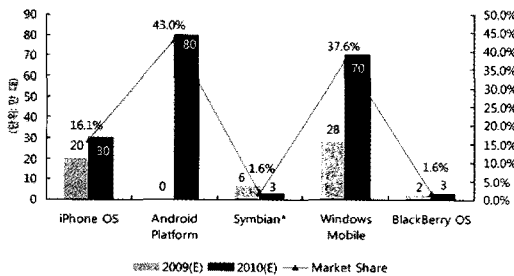
표 2. 국내 이동 통신사별 주요 스마트폰 플랫폼 환경

스마트폰 플랫폼	Window Mobile	Android	iOS
이동통신사업자	SKT/KTF/LGT	SKT	KTF
개발업체	Microsoft	Google	Apple
실행환경	C/C++/C#/Java	Java	Objective-C

표 3. 국내 이동 통신사별 주요 모바일 플랫폼 환경

모바일 플랫폼	WIPI	GNEX	SK-VM	BREW	MAP	자바스테이션
이동통신사업자	SKT/KTF/LGT	SKT	SKT	KTF	KTF	LGT
개발업체	KWISF	신지소프트	XCE	퀄컴	모빌탑	SUN(LG)
실행환경	C, Java	모바일 C	Java	ANSI C/C++	ANSI C	Java

국내 스마트폰 시장 OS & Platform 전망



Source: ROA Group Korea 추정치
 Note: Symbian(Nokia Express Music), 법인용 스마트폰 포함

그림 5. 국내 스마트폰 시장 OS 및 플랫폼 전망

수 있는 좋은 기회를 맞고 있다. 그러나 작은 시장 규모와 제한된 CP 풀(2008년 기준 모바일 게임 등급 심사 CP는 100여개정도) 및 다양한 모바일 및 스마트폰 플랫폼의 등장으로 변화에 빠르게 대처해야 하는 매우 어려운 환경을 맞이하고 있다. 또한, 국내 기술도 휴대폰 단말기에만 집중되어 있지 다른 IPTV, PMP, PDA 등과 같은 다양한 컨버전스 플랫폼으로는 확산되지 못한 실정이다. 또한, 국내특허분석결과 기존의 모바일 콘텐츠를 스마트폰 콘텐츠로, 스마트폰 콘텐츠를 다른 스마트폰 콘텐츠로 자동변환해주는 특허는 아직 등록된 사례가 없는 것으로 나타나고 있어 게임 산업의 경쟁력 확보를 위해서는 이와 같은 기술의 확

보가 시급한 실정이다.

이와 같은 국내 스마트폰용 플랫폼 기술의 강점과 약점, 대내외 환경의 유리한 점과 불리한 점을 SWOT 분석을 통해 살펴보면 표 4와 같다.

(5) 국내외 표준화 현황

국내외에서 기존의 모바일 핸드폰이나 스마트폰에서 모바일 콘텐츠의 변환 솔루션과 관련하여 표준화가 추진되거나 제정된 사례는 없다. 그것은 모바일 콘텐츠의 변환 솔루션 자체가 표준화를 추진하기에 적합하지 않은 형태이므로 주로 변환을 위한 방법론 적으로 WIPI와 같은 목적 플랫폼에 독립적인 프로그래밍 언어의 통합이나, 가상기계 포팅 등의 형태가 추진되고 있다. 앞서 언급된 사례와 같이, 'MIDP on BREW'나 'GNEX on WIPI', 'WIPI on Symbian', 그리고, WIPI2.0에서 MIDP2.0을 포함하는 등의 사례가 그 예라 하겠다. 모바일 게임 시장의 성숙과 함께, 기술적으로는 3D 및 여러 가지 분야에서 새로운 표준들이 지속적으로 정의 및 사용되고 있으나, 지역적 특성과 시장 원리에 입각한 비표준의 지배가 약해지기까지는 상당한 시간을 필요로 할 것으로 예상된다. 국내만하더라도 WIPI가 통합 플랫폼으로 발표된 지 4년이 넘은 지금에도, 3사의 확장 API가 상이

표 4. 국내 스마트폰용 플랫폼 기술의 SWOT 분석

강점(Strength)	약점(Weakness)
다양한 형태의 플랫폼 상용화 경험이 가장 큰 강점	개발자 커뮤니티가 활성화되어 있지 않고 플랫폼 분야에 대한 지속적인 R&D가 제한적임
기회(Opportunities)	위협(Threats)
국내 단말 제조사의 세계 시장 점유율 확대는 단말 기반 플랫폼 제작의 호기를 제공함	국내 스마트폰 출하량이 미미하여 신규 단말 및 단말 플랫폼의 수요 시장이 작음

하게 발전되어가고 있어, 통합 플랫폼이라는 이름을 무색하게 하고 있다[25-39,45].

1.2 연구개발의 필요성

현재 국내에서는 기존의 피쳐폰 모바일 플랫폼으로 GNEX, WIPI, MIDP, BREW 등이 사용되고 있으며, 스마트폰 플랫폼으로 Android, iPhone (iOS), Window Mobile 등이 공존하고 있다. 이동통신사별로 각기 서로 다른 플랫폼을 채택하여 사용함으로써 콘텐츠 실행 환경이 서로 달라 모바일 콘텐츠 개발의 커다란 장애 요소로 작용하고 있는 실정이다. 개발자는 하나의 콘텐츠를 서비스 하기 위해 각각의 플랫폼의 특성에 맞추어 다양한 버전을 모두 지원해야 한다. 즉, 우수한 모바일 콘텐츠를 개발하는데 시간과 비용이 투자되기 보다는 하나의 콘텐츠를 서비스하기 위하여 동일 콘텐츠를 중복 개발하거나, 타 플랫폼으로 이식하기 위한 변환 작업에 시간과 비용이 중복 투자되고 있다. 이는 개발 인력의 분산, 상품 기획력 둔화 등으로 이어져 결국에는 모바일 콘텐츠의 질적 하락을 가져오고 있으며, 모바일 산업의 경쟁력을 약화시키고 있다.

본 논문에서는 다양한 스마트폰 플랫폼으로 콘텐츠를 서비스하기 위한 개발 과정 또는 변환 과정에서의 시간과 비용의 소모를 최소화하기 위한 모바일 콘텐츠 자동 변환기 시스템을 개발하는데 그 목적이 있다. 스마트폰 플랫폼을 위한 모바일

콘텐츠 자동 변환기 시스템은 소스 형태의 콘텐츠를 목적 플랫폼의 소스 형태로 자동 변환하여 개발자가 하나의 모바일 콘텐츠를 개발하였을 때 이 시스템을 통해 개발된 콘텐츠를 다른 플랫폼으로 이식하는 작업을 단기간 내에 완료할 수 있도록 할 수 있다. 또한, 기존 콘텐츠를 자동 변환하여 다른 플랫폼에 서비스함으로써 콘텐츠의 재사용성(OSMU: One Source Multi Use)을 높이고, 신규 콘텐츠의 생산성을 높여 사용자에게는 다양한 모바일 게임 콘텐츠를 제공할 수 있도록 지원한다. 따라서, 본 연구에서 개발할 콘텐츠 자동 변환기는 스마트폰 플랫폼뿐만 아니라 현재 이슈가 되고 있는 IPTV, PMP, PDA 등과 같은 컨버전스 플랫폼 환경에서도 꼭 필요한 소프트웨어 기술이다. 그리고 또한, 국내외 특허분석결과 기존의 모바일 콘텐츠를 스마트폰 콘텐츠로, 스마트폰 콘텐츠를 다른 스마트폰 콘텐츠로 자동변환해주는 특허는 아직 등록된 사례가 없는 것으로 나타나고 있어 게임 산업의 경쟁력 확보를 위해서는 이와 같은 기술의 확보가 시급한 실정이다[25-39,45]. 그림 6은 콘텐츠 변환기를 이용하여 기존의 위피 게임을 다양한 스마트폰 플랫폼 콘텐츠로 변환하여 실행하는 예이다.

1.3 모바일 콘텐츠의 멀티소스 변환모델

기존의 콘텐츠 변환 기술로는 변환을 수동으로 처리해야 하기 때문에 1:1 변환만이 가능하다. 그

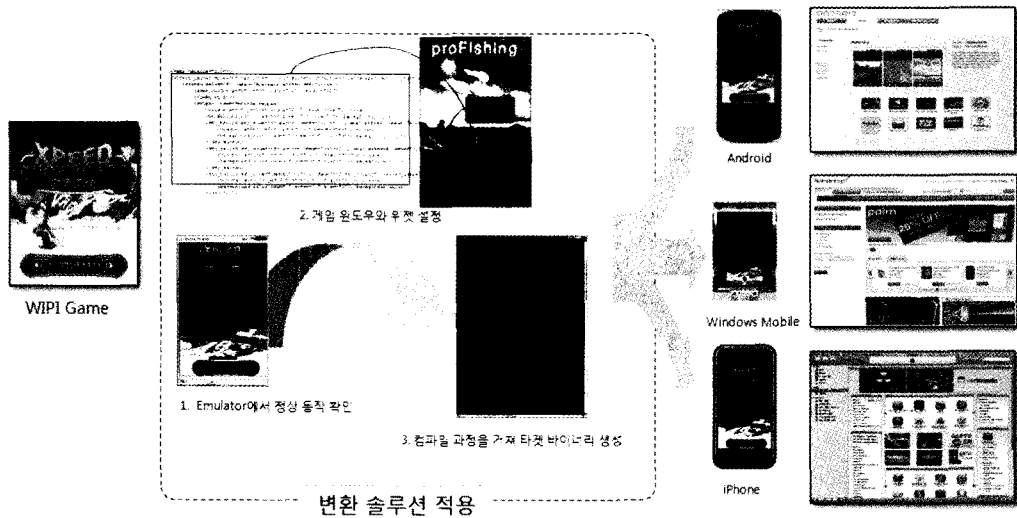


그림 6. 스마트폰 플랫폼을 위한 WIPI 게임의 콘텐츠 변환 예

렇지만 제안한 컴파일러 제작 기술을 콘텐츠 변환 솔루션에 응용하여 자동으로 변환하면 각각 다른 언어로의 변환이 가능하기 때문에 1:n 또는 n:n 변환이 가능하다. 또한, 스마트폰 플랫폼 뿐만 아니라 IPTV나, PMP, PDA와 같은 컨버전스 플랫폼 환경을 위한 콘텐츠 변환이 가능하다.

현재 출시되고 있는 스마트폰들의 플랫폼이 모두 다르고 표준화되어 있지 않으며 다양한 프로세서에 대한 독자적인 프로그램 개발 환경을 갖고 있기 때문에 플랫폼마다 프로그램을 개발해야 하는 속성을 가지고 있으며 같은 작업을 하는 동일한 프로그램에 대하여 플랫폼마다 프로그램을 다

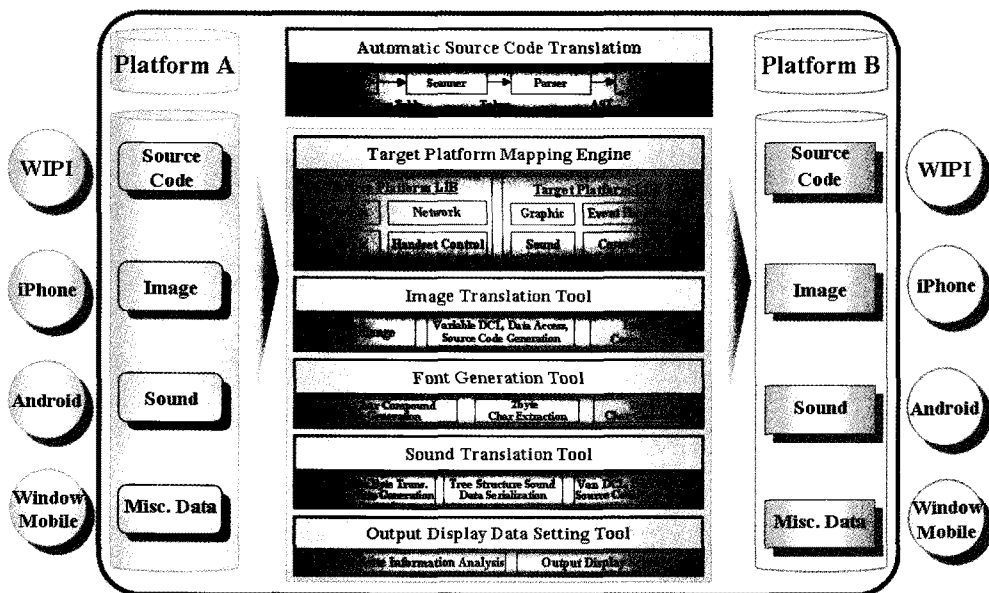


그림 7. 멀티소스 변환 모델의 시스템 구조

시 개발해야 하는 단점을 내포하고 있다. 이러한 단점을 극복하기 위한 방법으로 콘텐츠를 재사용할 수 있는 콘텐츠 변환 기술이 효과적이며 OSMU(One Source Multi Use)를 구현할 수 있는 방법 중 하나이다[29-39,45].

본 논문에서 제안한 세부내용의 개발된 연구결과를 통합함으로써 다양한 플랫폼을 위한 콘텐츠 자동 변환기가 실질적으로 개발되어질 수 있다. 이러한 기술은 하나의 언어로 제작된 콘텐츠를 최소한의 노력으로 다양한 스마트폰 플랫폼에서 실행할 수 있는 기술로 게임 콘텐츠의 OSMU를 구현하는 기술이며 디지털 콘텐츠를 위시한 지식산업의 발전과 관련 기업의 경쟁력을 향상시킬 수 있는 새로운 기술이다

2. 피쳐폰 플랫폼과 스마트폰 플랫폼

2.1 피쳐폰 플랫폼

(1) WIPI 플랫폼

WIPI(Wireless Internet Platform for Interoperability)는 한국 무선 인터넷 표준화 포럼(KWISF)에 의해 제정되고, 한국정보통신기술협회(TTA)에 의해 TTAS.KO-60.0036으로 채택된 이동통신 단말기용 응용프로그램 실행 환경을 표준화한 규격이다. 이동통신 사업자들이 각기 다른 플랫폼을 사용함으로써 단말기 제조사와 콘텐츠 업체들의 개발 부담이 크다는 점 등을 들어 국내 무선인터넷 플랫폼 표준화의 필요성이 대두되면서 제정한 국내 표준 규격이다[20].

WIPI는 응용프로그램 개발 언어로 네이티브(Native) 방식의 C와 자바를 모두 지원한다. 자바의 경우 AOTC(Ahead Of Time Compiler)를 통해 C로 변환하여 다시 네이티브 방식으로 실행된다. 위피 규격은 크게 HAL(Handset Adaptation

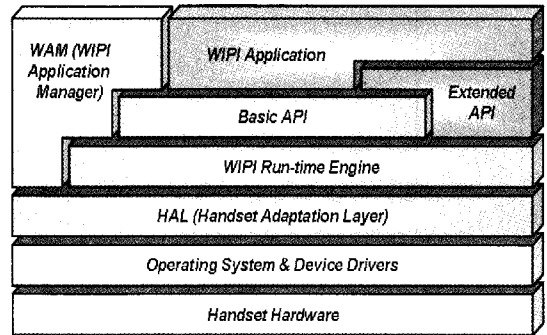


그림 8. WIPI 플랫폼의 구조

Layer)과 기본 API(Application Programming Interface)로 구성된다. HAL은 플랫폼 이식성을 높이기 위한 표준화된 하드웨어 추상화 계층이다. 기본 API는 표준화된 플랫폼 호환성을 제공해 다양한 응용 프로그램 개발을 촉진하기 위한 기본 API 모음으로 C와 자바 API로 구성되어 있다.

(2) GNEX 플랫폼

GNEX 가상기계(virtual machine)는 GNEX 응용 프로그램을 해석하고 실행하는 역할을 한다. GNEX 커널은 다양한 시스템 인터페이스를 제공하며, 메모리 관리자 탑재로 GNEX 시스템을 보고하고 응용 프로그램의 크기 및 힙(heap) 메모리 제약 등을 해소하는 역할을 한다. 이벤트 핸들러는 플랫폼의 이벤트를 받아 GNEX 이벤트로 변환

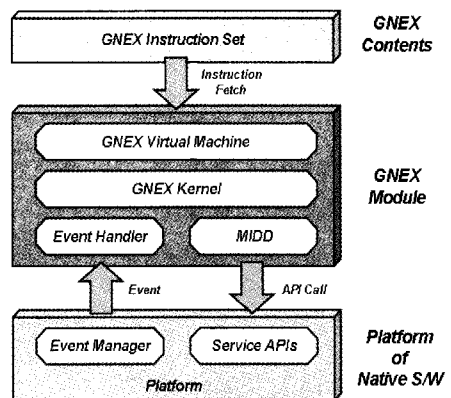


그림 9. GNEX의 구조

하고 각 이벤트에 대응되는 알고리즘을 호출하여 처리한다. MIDD(Mobile Interface Device Driver)는 사운드 재생, LCD 출력 등 단말기의 하드웨어 관련 기능을 단말기 플랫폼에서 제공하는 API를 이용하여 구현한 것이며, 플랫폼 API의 호출과 실행 결과를 처리한다.

GNEX 응용 프로그램은 ANSI C 언어를 기반으로 한 모바일 C 언어로 개발한다. GNEX는 VDI(Variable Depth Image)라는 단말기 전용으로 설계된 이미지 형식을 사용한다. VDI는 픽셀당 할당되는 비트 수를 가변적으로 정의하여 사용하는 일종의 비트맵 형식의 이미지 규격이다. GNEX 응용 프로그램에서는 BMP와 같은 이미지 파일을 GNEX SDK에서 제공하는 이미지 도구를 통해 VDI 형식으로 변환하여 모바일 C 소스 코드에 포함시킨다. 사운드 리소스 역시 GNEX 응용 프로그램에서 사용할 수 있도록 GNEX 규격의 사운드 파일로 변환하여 모바일 C 소스 코드에 포함하여 사용한다[22-24].

(3) MIDP 플랫폼

자바는 인터넷 분산 환경 시스템에서 효과적으로 응용 프로그램을 작성할 수 있도록 개발된 프로그래밍 언어로서, 객체지향 패러다임의 특성 및 다양한 개발 환경을 지원한다. 휴대폰, PDA 등과 같은 단말기를 목적으로 하는 자바 플랫폼을 J2ME(Java 2 Micro Edition)이라 한다. J2ME의 구조를 살펴보면 단말기의 운영체제 위에 가상기계(Virtual Machine)가 존재하고 그 위에 컨피그레이션(Configuration)과 프로파일(Profile)이 수직적인 구조로 위치하고 있다. 컨피그레이션이란 메모리 용량, 프로세서 속도 등 단말기에서 공통적으로 필요한 최소한의 가상 머신 기능과 라이브러리에 대하여 정의한 것이고, 프로파일은 그 상위의 클래스 라이브러리에 대한 명세를 의미한다.

단말기용 자바플랫폼 CLDC(Connected Limited Device Configuration)/MIDP(Mobile Information Device Profile)는 KVM(Kilobyte Virtual Machine)을 기본 가상기계로 하는 CLDC와 무선 단말기 시장을 겨냥해서 CLDC를 기반으로 한 클래스 라이브러리에 대한 명세인 MIDP로 구성된다. MIDP 어플리케이션을 MIDlet이라 하며, 자바 애플릿과 흡사하다. CLDC는 KVM을 기본 가상기계로 채택하고 있는데, KVM은 단말기와 같이 메모리가 작고 CPU 속도가 느린 환경을 위하여 만든 작은 크기의 가상기계이다. CLDC는 KVM 규격을 기본 가상기계로 채택하고 여기에 코어 API, 네트워킹과 입출력, 보안과 국제화 등에 대한 정의를 포함한 J2ME 컨피그레이션이다.

MIDP는 이동 단말기에 적합하도록 CLDC의 상위에서 동작하는 자바 클래스 라이브러리에 대한 명세로 기본 CLDC 명세서를 확장하고 구체화시킨 것이다. MIDP 내에 포함된 API는 어플리케이션, 사용자 인터페이스, 영속적 저장 공간, 네트워킹, 타이머 등이 있다. 그림 10은 MIDP 플랫폼의 구조이다.

이동통신 사업자마다 서비스의 차별화를 위한 기능 추가로 OEM 명세 클래스(OEM-Specific Classes)가 CLDC 상위에서 제공될 수 있으며, 이를 이용해 무선 네트워크 특성 및 단말기의 특성

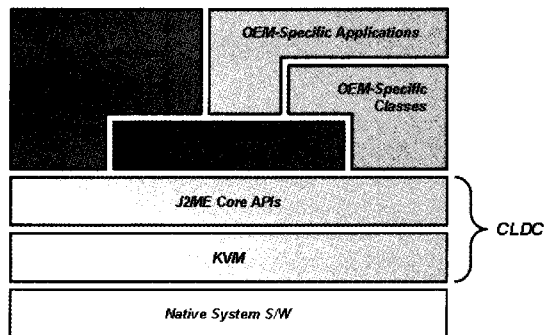


그림 10. MIDP 플랫폼의 구조

을 이용할 수 있는 OEM 기반의 서비스를 제공할 수 있다[5-7,21].

2.2 스마트폰 플랫폼

(1) iOS 플랫폼

iPhone 플랫폼인 iOS(iPhone Operating System)는 애플의 스마트폰인 아이폰과 디지털 미디어 재생기인 아이팟 터치, 태블릿형 컴퓨터인 아이패드에 내장되어 있는 운영체제이다. 실제로는 Mac OS X를 기반으로 만들어져 있으며 Mac OS X의 기본 구조와 비슷하다. iOS는 하드웨어와 화면에 보이는 어플리케이션의 중간 역할을 한다. 어플리케이션은 직접 하드웨어와 연결되지 않으며 시스템 인터페이스를 통해서 연결된다.

iOS는 코어 OS 계층(core OS layer), 코어 서비스 계층(core services layer), 미디어 계층(media layer), 코코아 터치 계층(cocoa touch layer)의 네 개의 소프트웨어 계층으로 구성되어있다. 그림 11은 iOS의 구조를 나타낸 것이다[11-14].

코어 OS와 코어 서비스 계층은 기본적인 시스템 API등을 제공한다. iOS는 Mac OS X와 같은 유닉스 기반으로 POSIX 표준을 준수하는 API들을 제공한다. 네트워크 소켓에 관련된 부분이라든가 스레드 같은 부분은 POSIX 표준으로 바로 작성할 수 있다. 미디어 계층에서는 C 언어와 Objective-C 언어를 섞어서 사용하는 프레임워크들이 제공된다. 3D 기능을 사용할 수 있는

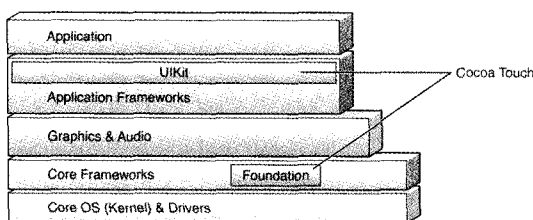


그림 11. iOS 플랫폼의 구조

OpenGL ES부터 이를 코코아에서 편하게 쓸 수 있도록 도와주는 코어 애니메이션 프레임워크, 그리고 소리 재생과 관련된 코어 오디오 프레임워크 등이 지원된다. 코코아 터치 계층은 아이폰 및 아이팟 터치용 API로서, 4개의 추상화 계층 중 가장 상위 계층에 존재하며, 개발자로 하여금 하위 계층을 자세히 알 수고를 덜어준다. GUI를 구현하고 이벤트-구동(event-driven) 기법을 쓰는 아이폰 및 아이팟 터치용 응용 소프트웨어는 보통 코코아 터치 계층에 기반하여 작성된다. 코코아 터치는 iOS에 대한 일종의 추상화 계층을 제공한다.

(2) 안드로이드 플랫폼

안드로이드(Android) 플랫폼은 구글에서 개발한 운영체제, 미들웨어, 응용 프로그램을 포함하는 모바일 기기에 최적화된 플랫폼이다. 안드로이드 플랫폼은 오픈 소스 정책을 채택하였으며, 리눅스 커널, 라이브러리, 런타임, 어플리케이션 프레임워크, 어플리케이션으로 구성되어 있다[5,6].

그림 12는 안드로이드 플랫폼의 계층 구조와 구성요소를 도시한 것이다. 리눅스 커널은 보안, 메모리 관리, 프로세스 관리, 네트워크 스택, 드라이버 모델과 같은 리눅스 버전 2.6의 핵심 시스템

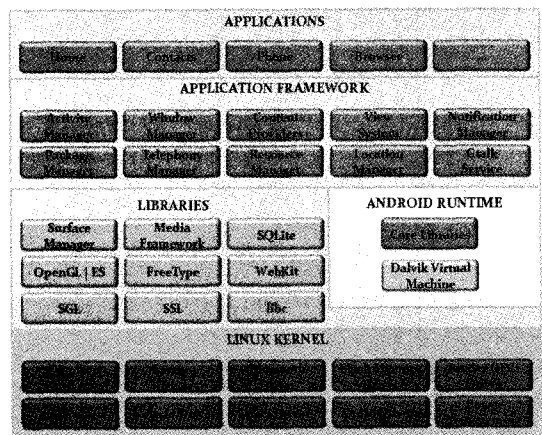


그림 12. 안드로이드 플랫폼의 계층 구조

서비스를 이용하며, 커널은 하드웨어와 소프트웨어 간의 추상 계층으로 동작한다. 라이브러리는 C와 C++로 구성되었으며 C 시스템 라이브러리, 미디어 라이브러리, 3D 라이브러리 등을 제공한다. 애플리케이션 프레임워크는 자바로 구성된 패키지 컴포넌트이며, 애플리케이션은 이 프레임워크의 패키지를 사용하여 작성할 수 있다[8-10].

안드로이드 플랫폼의 모든 응용 프로그램은 자바로 작성되며, 작성된 응용 프로그램은 자바 컴파일러에 의해 클래스 파일로 변환된다. 생성된 클래스 파일은 실행에 앞서 DEX(Dalvik Executable File) 형식으로 다시 변환되며, dalvik 가상기계에 의해 실행된다. DEX 파일은 저장 공간과 메모리 사이의 효과적인 매핑 정의에 최적화된 포맷이며, dalvik 가상기계는 레지스터 기반의 가상기계로 제한된 메모리에 최적화된 특징을 가진다.

(3) 윈도우 모바일 플랫폼

윈도우 모바일(Windows Mobile, WM, 추후 윈도우 폰(Windows Phone)으로 브랜드 명칭 변경)은 PDA 및 스마트폰에 사용하는 운영 체제이다. 이전에는 포켓 PC라고 불렸다. 이 운영체제는 마이크로소프트사에서 내놓은 모바일 운영체제로 임베디드용 운영 체제인 윈도우 CE를 기반으로 하고 있고, 모바일 환경에 적합한 새로운 터치식 사용자 인터페이스를 추가하여 개발되고 있으며, 윈도우 CE의 기본적인 기능에 휴대전화 기능이 추가되어 있어서 현재 삼성전자, LG전자, HTC, 모토로라, 노키아 등의 기업체에서 생산하는 다수의 PDA폰에서 윈도 모바일을 탑재하고 있다.

윈도우 모바일 6 버전은 모바일 장치를 위한 플랫폼으로 윈도우 CE 5.0을 기반으로 제작 되었다. 그리고 스마트폰과 PDA와 같은 하드웨어를 지원한다. 마이크로소프트 비주얼 스튜디오 2005와 윈도우 모바일 SDK로 비주얼 C++와 비주얼

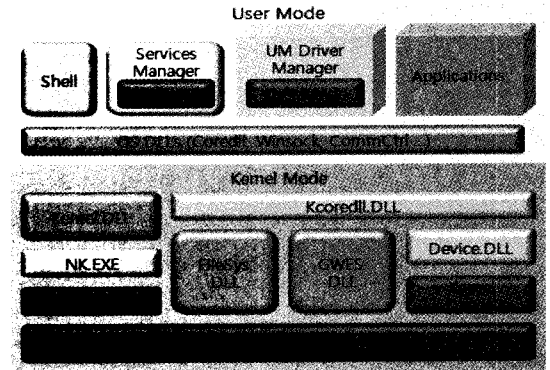


그림 13. 윈도우 모바일 구조

C#, 비주얼베이직 .NET 코드로 윈도우 모바일 플랫폼을 위한 소프트웨어 개발을 할 수 있다. 그림 13은 윈도우 모바일의 구조를 나타낸 것이다.

윈도우 모바일 6.5 버전은 윈도우 모바일 장치에 윈도우 데스크탑의 계열을 적용한 것이다. 여기선 터치스크린에 맞게 UI가 상당수 변경되었으며, 기존에 지원했던 클래식 포켓 PC 버전과 잘 사용되지 않는 해상도의 버전이 삭제되었고, 윈도우 모바일 6.1.4에서보다 더 강화되고 간편해진 인터넷 익스플로러 모바일 6가 기본 내장되어 있다. 윈도우 모바일은 윈도우 임베디드 CE 5.2 기반이며 .NET 컴팩트 프레임워크를 지원한다[15].

윈도우 모바일 플랫폼을 사용하면 획기적인 모바일 어플리케이션을 제작 할 수 있다. 이 플랫폼은 데이터 연결과 강화된 보안을 제공한다. 블루투스와 POOM(Pocket Outlook Object Model)과 같은 다양한 API를 제공한다. Native 코드(CPP), managed 코드(C#), 모바일 웹 개발, 멀티 스레딩과 같은 장치 자원과 같은 프로그래밍 모델들을 광범위하게 포함한다. 또한, 윈도우 개발 환경과 유사한 이점으로 개발시간과 비용을 줄일 수 있다.

3. 모바일 콘텐츠 자동 변환기 시스템

스마트폰 플랫폼을 위한 모바일 콘텐츠 자동

변환기 시스템을 설계하고 구현하기 위해, 스마트폰을 위한 모바일 콘텐츠 자동 변환기 시스템 모델을 제안하고, 그 모델로부터 야기되는 다음과 같은 다섯 가지 분야를 개발한다: (1) 입력받은 콘텐츠의 소스 코드에서 리소스 데이터를 분리하고 사용된 폰트를 추출하는 콘텐츠 분석기의 개발 (2) 콘텐츠 제작에 사용된 소스 코드를 컴파일러 제작 기술을 이용하여 목적 플랫폼의 소스 코드로 자동 번역해주는 소스 코드 자동 번역기의 개발 (3) 콘텐츠에 사용된 이미지, 사운드와 같은 리소스 데이터를 목적 플랫폼에 맞게 변환해주는 리소스 변환기의 개발 (4) 콘텐츠에 사용된 폰트를 목적 플랫폼에서 사용할 수 있도록 동일한 규격의 폰트를 생성하는 폰트 생성기의 개발 (5) 콘텐츠의 실행환경과 API 함수를 목적 플랫폼에 맞게 구축해주는 플랫폼 매핑 엔진(API 변환기)의 개발

본 논문에서 제안하고 개발하고자 하는 스마트폰을 위한 모바일 콘텐츠 자동 변환기 시스템 모델은 그림 14와 같다[32-34,45].

3.1 콘텐츠 분석기

콘텐츠 분석기(content analyzer)는 입력받은 콘텐츠를 분석하여 콘텐츠에 내재된 리소스 데이

터와 소스 코드를 분리하여 출력하는 시스템이다. 기존의 GNEX와 같은 플랫폼은 파일 시스템을 지원하지 않고 리소스 데이터와 사용자 데이터를 텍스트 형태로 소스 코드에 입력하여 사용한다. 즉, 콘텐츠 분석기는 이미지나 사운드 타입의 변수 형태인 리소스 데이터를 각 변수별로 분리하고 구조체에 정보를 담아 리소스 목록을 생성하고 실제 이미지나 사운드 리소스 데이터 인지, 맵 타일(map tile) 등의 사용자 데이터 인지 구분하여 리소스 변환기에 전달한다. 또한, 콘텐츠에 사용된 문자를 추출하여 문자 목록을 만들어 목적 플랫폼에서 동일한 문자를 사용할 수 있도록 폰트를 생성할 수 있게 한다. 그리고 리소스 부분을 제외한 나머지 소스 코드 부분은 소스 번역기에 전달하여 목적 플랫폼의 소스 언어로 자동번역하게 한다. 그림 15는 콘텐츠 분석기의 시스템 모델이다[32,34].

3.2 소스 코드 자동 번역기

기존의 콘텐츠 변환기가 소스 코드의 변환을 위해 문장의 단어에 대한 스트링 패턴 매칭 방식을 사용하거나 프로그래머가 직접 변환을 하는 수동적인 방법을 사용하지만, 본 연구에서 제안한

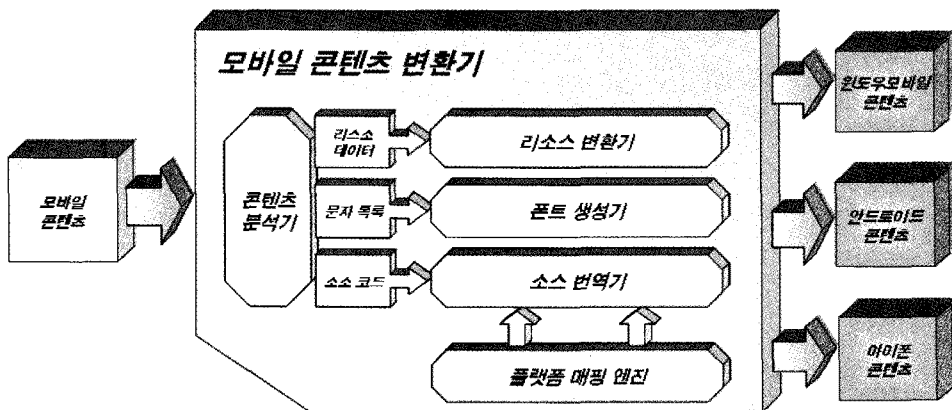


그림 14. 스마트폰을 위한 모바일 콘텐츠 자동 변환기 시스템 모델

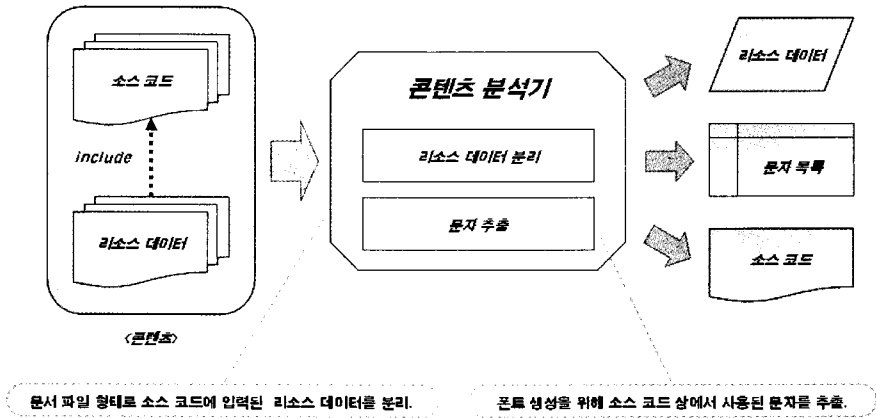


그림 15. 콘텐츠 분석기의 시스템 모델

방법은 시스템 소프트웨어인 컴파일러 제작 기술을 사용하여 소스 코드를 목적 플랫폼의 언어로 체계적으로 자동 변환하는 방법을 사용한다. 따라서 소스 자체를 문법적 단위로 접근 하게 되어 모든 경우에 맞추어 변환이 가능하게 된다. 또한 그 문법적인 상태에서 다른 언어, 예를 들어 C에서 Java, Java에서 C로의 변환도 쉽게 가능하다.

소스 코드 자동 번역기(language translator)는 컴파일러 기술을 이용하여 입력받은 콘텐츠의 소스 코드를 목적 플랫폼의 소스 코드로 정형한 방법을 사용하여 자동 변환해주는 시스템이다. 이를 위해 먼저, 소스 언어의 문법을 컴파일러의 LALR(1) 문법으로 표기한 후에 PGS(Parser Generating System)를 이용하여 파싱테이블을 생성하고, 전처리기(preprocessor), 어휘 분석기(lexical analyzer), 구문 분석기(syntax analyzer) 및 소스 번역기(source translator)를 구현한다.

전처리기는 #include, #define 등의 지시문을 처리하여 입력받은 소스 코드를 확장하고, 어휘 분석기는 확장된 소스 코드를 읽어 들여 문법적으로 의미를 갖는 최소의 단위인 토큰(token)을 생성한다. 구문 분석기는 구문과 문법 분석을 위해 어휘 분석기가 생성한 토큰들을 받아 PGS에서 생성한

파싱테이블을 이용하여 소스 코드에 대한 에러를 체크하고 올바른 문장에 대해서는 구문 구조를 트리 형태로 생성한다. 본 번역기에서는 대부분의 컴파일러에서 사용하고 있는 추상구문트리(AST: Abstract Syntax Tree)를 사용할 예정이다. 언어 번역기는 구문 분석기가 생성한 AST를 입력으로 받아 트리를 탐색하면서 소스 언어와 의미적으로 등등한 목적 플랫폼의 언어로 자동 변환한다. 이와 같은 자동 변환 방법은 국내에서 최초로 시도하는 방법으로 그동안 게임 솔루션 업체들이 사용한 수동적인 방법 즉, 프로그래머가 개입하여 각 문장을 비교하여 그 문장과 일치하는 목적 언어로 변환하는 방법과는 차별적인 방법인 것이다[1-4, 16-19, 29-34]. 그림 16은 소스 코드 자동 번역기의 시스템 모델을 나타낸 것이다.

3.3 리소스 변환기

리소스 변환기(resource converter)는 입력 콘텐츠에 사용된 텍스트 또는 바이너리 형태의 리소스 데이터를 목적 플랫폼의 파일 시스템에서 사용할 수 있도록 이미지 데이터와 사운드 데이터, 그리고 사용자 데이터로 변환하여 각 파일로 변환하는 시스템이다. 이를 위해 각 플랫폼에서 사용하

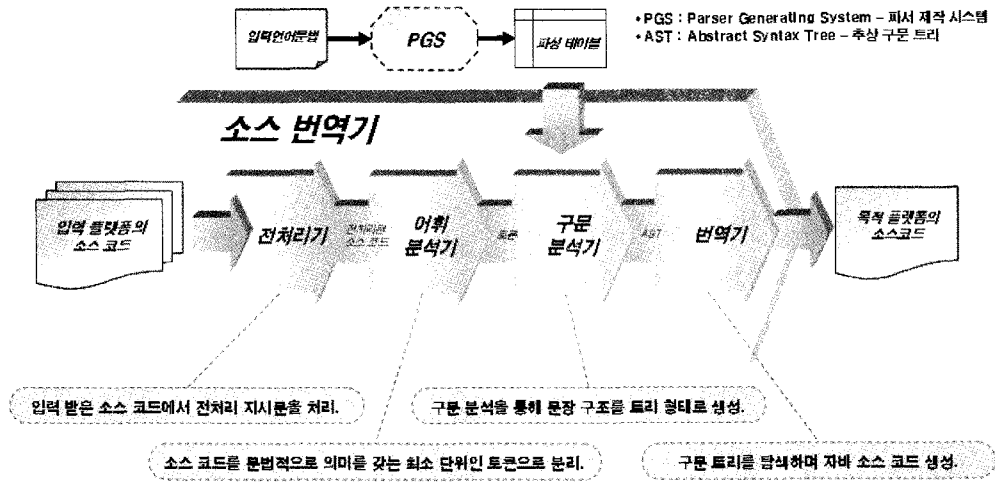


그림 16. 소스 코드 자동번역기의 시스템 모델

는 이미지 파일 형식(자체 이미지 형식-VDI, BMP, PNG, JPEG GIF 등), 사운드 형식(MMF, MP3, MIDI 등), 사용자 데이터를 조사하고 목적 플랫폼에서 사용할 수 있도록 변환해 주어야 한다 [29-34].

즉, 기존의 모바일 플랫폼에서 이미지 파일 형식으로 지원하는 BMP나 PNG, JPEG 등과 사운드 형식으로 지원하는 야마하 형식(ma1, ma2,...) wav, mp3 등은 스마트폰 플랫폼인 안드로이드나 아이폰에서 지원하고 있기 때문에 변환 없이 텍스트 형태에서 바이너리 파일이나 사운드 파일 형태

로 출력하면 된다. 그리고 어떤 리소스가 사용되었는지를 나타내는 관리파일만 작성해주면 된다. 그리고 자체 형식을 사용한 경우에는 그 기능을 지원하기 위해 형식은 유지하고 텍스트 형태에서 해당 플랫폼의 형식으로 변환하고 해당기능을 지원하는 API 함수를 통해 사용할 수 있게 해주면 된다. 또한, 텍스트 형태의 사용자 데이터도 해당 플랫폼의 파일 시스템을 통해 접근 가능하도록 바이너리 파일 형태로 출력해야 한다. 그림 17은 리소스 변환기의 시스템 모델이다.

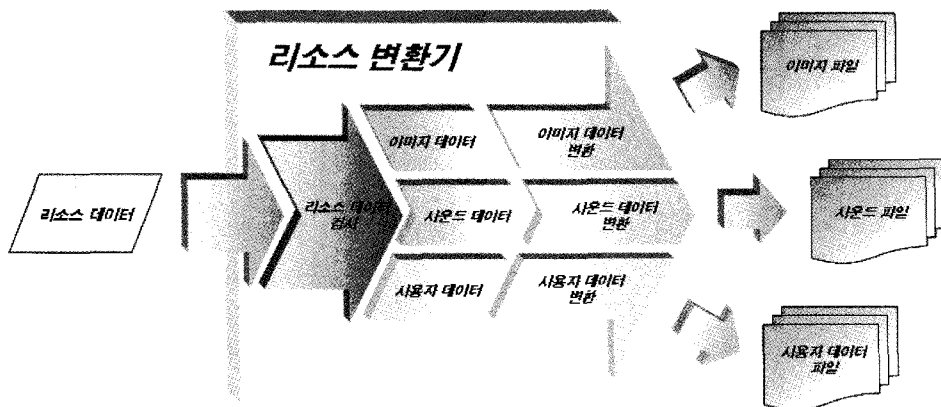


그림 17. 리소스 변환기의 시스템 모델

3.4 폰트 생성기

폰트 생성기(font generator)는 입력 콘텐츠에 사용된 폰트와 동일한 규격의 폰트를 생성하여 목적 플랫폼에서 사용할 수 있도록 제공하는 시스템이다. 플랫폼마다 폰트 규격이 서로 다르기 때문에 입력 플랫폼과 동일한 폰트를 생성해서 제공하지 않고서는 문자 출력이 있어서 동일한 결과를 얻을 수 없다. 본 연구에서 구현할 폰트 생성기는 입력 플랫폼의 폰트와 글씨체까지 같지는 않지만 동일한 크기의 폰트를 생성하여 입력 플랫폼의 폰트 규격을 목적 플랫폼에서 사용할 수 있도록 할 예정이다. 기존의 모바일 플랫폼인 GNEX에서는 자체 폰트를 사용하여 폰트의 형태 및 규격이 명확하게 규정되어 있다. 하지만 WIPI, Android, iPhone과 같은 플랫폼에서는 각각 지원하는 폰트의 규격이 다르며 이동통신사, 단말기 제조사에 따라 폰트의 형태와 규격이 제각각이다.

폰트 생성기는 폰트를 생성하고자 하는 문자의 비트맵 이미지를 만들고 그 비트맵 이미지의 픽셀 정보를 추출하여 폰트를 화면에 출력하기 위해 필요한 데이터를 생성한다. 영어, 숫자와 같은 1바이트 문자의 경우 “SMALL”, “MEDIUM”, “LARGE”, “HUGE” 규격에 해당하는 폰트 이미

지 데이터를 생성하고 한글, 특수 문자와 같은 2바이트 문자는 “LARGE”, “HUGE” 규격에 해당하는 폰트 이미지 데이터를 생성한다. 다음 그림 18은 폰트 생성기의 시스템 모델이다.

폰트 데이터의 1비트는 화면에 출력할 폰트 이미지의 한 픽셀을 나타낸다. 문자 출력 메시드는 값이 0인 비트에 맞춰 화면에 문자를 출력하도록 한다. 폰트 이미지의 각 라인은 바이트 단위로 구성된다. 1바이트 문자의 폰트 데이터는 한 라인이 1바이트(8비트)로 구성되고, 2바이트 문자의 폰트 데이터는 한 라인이 2바이트(16비트)로 구성된다. 한 라인을 구성하고 남은 비트 부분은 사용하지 않는다[29-34].

3.5 플랫폼 매핑 엔진

소스 번역기는 소스 플랫폼과 목적 플랫폼 간의 사용 언어를 자동으로 번역해주는 기능을 할 뿐, 실행할 목적 플랫폼의 응용 프로그램을 생성하지는 못한다. 즉, 소스 번역기를 통해 번역된 소스 코드는 목적 플랫폼에서 바로 실행되지 않는다. 소스 코드에서 사용된 API 함수, 시스템 변수, 이벤트 핸들러 등을 목적 플랫폼에서 지원하는 형태로 변환해주어야 실행이 가능해진다. 플랫폼



그림 18. 폰트 생성기의 시스템 모델

매핑 엔진은 이 작업을 자동으로 처리하는 역할을 한다.

즉, 플랫폼 매핑 엔진(platform mapping engine)은 입력 콘텐츠를 동일한 환경에서 실행하기 위해 입력 플랫폼의 실행환경을 목적플랫폼에 구축하고, 해당 API 함수를 제공하는 시스템이다. 이를 위해 API 함수, 시스템 변수, 이벤트 환경 등을 동일한 형태로 사용할 수 있도록 지원하여 변환된 소스 코드를 쉽게 이해하고 수정할 수 있도록 한다. 또한, 구동 환경의 동질성으로 인해 실행시의 신뢰성 및 안정성을 높일 수 있게 한다.

입력 플랫폼과 동일한 형태의 API 함수와 시스템 변수를 지원하기 위해 입력 플랫폼의 디스플레이, 그래픽, 사운드 출력, 이벤트, 진동, 데이터 저장 등의 시스템 환경과 유사한 환경을 구축하고 이를 바탕으로 시스템 변수 및 API 들을 목적 플랫폼에서 동일하게 동작하도록 구현한다. 또한, 입력 플랫폼의 이미지 형식 출력을 위한 환경, API 함수를 제공하고 이미지, 사운드 타입의 멀티미디어 자료형 등을 목적 플랫폼에서도 사용할 수 있도록 지원할 예정이다. 그리고 타이머와 시스템 API, 핸드셋 제어(handset control) API, 네

트워크 API, DB API 등도 동일한 동작을 하도록 구현한다. 이렇게 함으로써 번역된 소스 코드를 추가적으로 수정하지 않고 실행할 수 있도록 할 수 있다. 변환기에서 제공하는 API는 소스 플랫폼의 API와 유사한 형태로 동일한 기능을 수행하도록 래퍼 함수(Wrapper Function Set)를 사용하여 구현하였다. 이를 통하여 소스 플랫폼의 디스플레이, 그래픽, 사운드 출력, 이벤트, 진동, 데이터 저장 등의 시스템 환경과 유사한 환경을 제공한다. 이 환경을 바탕으로 소스 플랫폼의 시스템 변수 및 함수들을 목적 플랫폼에서 동일하게 동작하도록 구현하였다[29-34]. 그림 19는 플랫폼 매핑 엔진의 시스템 모델을 나타낸 것이다.

4. 모바일 콘텐츠 변환 결과

스마트 플랫폼을 위한 피쳐폰 모바일 콘텐츠의 자동변환을 위해 3장의 콘텐츠 분석, 리소스 변환, 소스 번역, 폰트 생성, 플랫폼 매핑 엔진을 통한 API 지원 등을 통해 콘텐츠 변환의 기능을 실험하기 위해 몇 가지 피쳐폰 콘텐츠를 변환하여 보았다. 다음은 그 실행 결과이다.

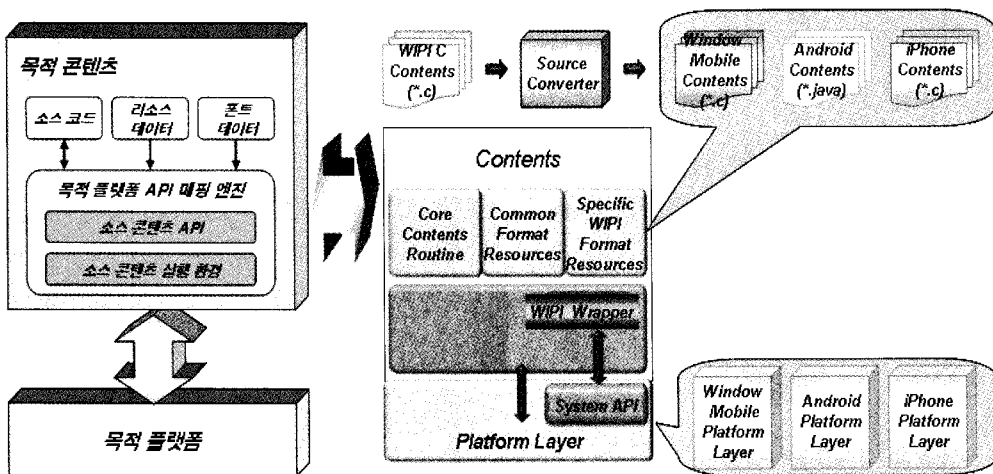


그림 19. 플랫폼 매핑 엔진의 시스템 모델

(1) 모바일 콘텐츠의 변환내용
(GNEX-to-WIPI, GNEX-to-MIDP)

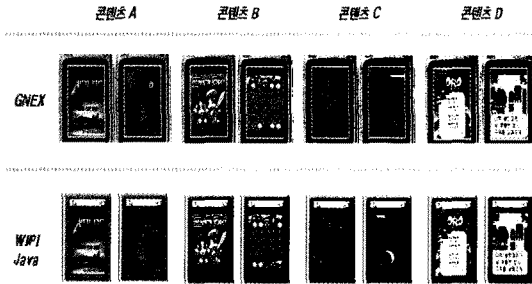


그림 20. GNEX/WIPI 플랫폼에서의 콘텐츠 실행 결과

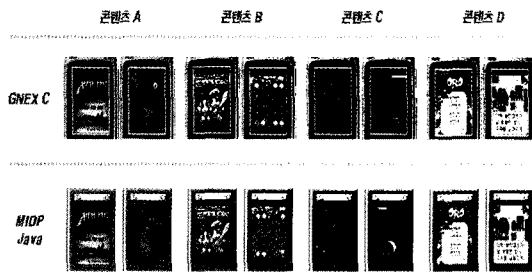


그림 21. GNEX/MIDP 플랫폼에서의 콘텐츠 실행 결과

(2) 모바일 콘텐츠의 변환내용
(GNEX-to-Android)

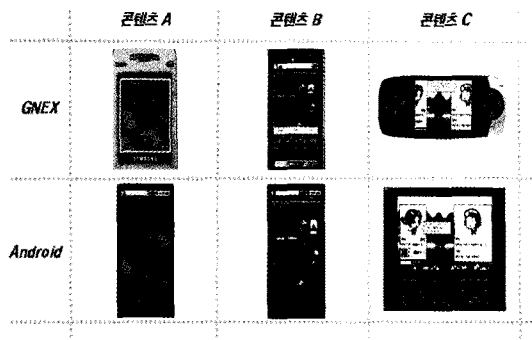


그림 22. GNEX/Android 플랫폼에서의 콘텐츠 실행결과

(3) 모바일 콘텐츠의 변환내용(GNEX-to-iOS)

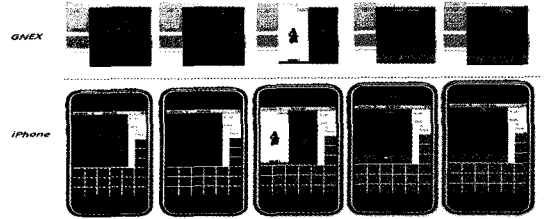


그림 23. GNEX/iOS 플랫폼에서의 콘텐츠 실행결과1

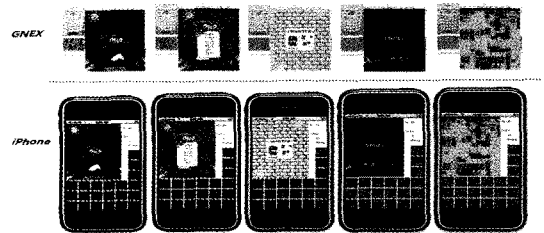


그림 24. GNEX/iOS 플랫폼에서의 콘텐츠 실행결과2

(4) 모바일 콘텐츠의 변환내용
(WIPI-to-Window Mobile)

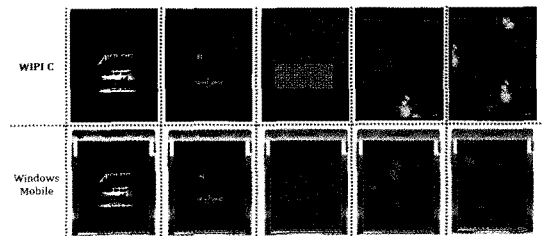


그림 25. WIPI/Window Mobile 플랫폼에서의 콘텐츠 실행결과1

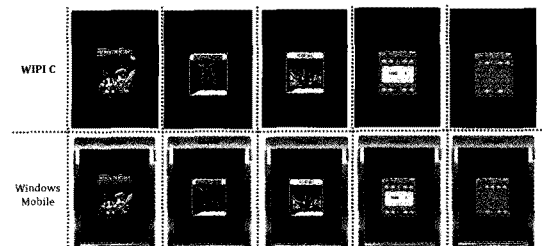


그림 26. WIPI/Window Mobile 플랫폼에서의 콘텐츠 실행결과2

5. 결론 및 향후 연구

스마트폰이 등장하면서 스마트폰 앱 콘텐츠 시장은 매년 높은 성장률의 시장 규모를 이어가며 스마트폰 시장 최고의 킬러 콘텐츠로 자리 잡았다. 하지만 스마트폰 플랫폼이 달라 하나의 스마트폰 앱 콘텐츠를 서비스하기 위하여 콘텐츠를 중복 개발하거나 변환해야 하는 문제점이 있다.

현재 출시되고 있는 스마트폰들의 플랫폼이 모두 다르고 표준화되어 있지 않으며 다양한 프로세서에 대한 독자적인 프로그램 개발 환경을 갖고 있기 때문에 플랫폼마다 프로그램을 개발해야 하는 속성을 가지고 있으며 같은 작업을 하는 동일한 프로그램에 대하여 플랫폼마다 프로그램을 다시 개발해야 하는 단점을 내포하고 있다. 이러한 단점을 극복하기 위한 방법으로 콘텐츠를 재사용할 수 있는 콘텐츠 변환 기술이 효과적이며 OSMU (One Source Multi Use)를 구현할 수 있는 방법 중 하나이다.

본 논문에서 제안한 모바일 콘텐츠 자동 변환기 시스템은 이런 문제를 해결할 수 있는 하나의 방법이다. 본 변환기는 기존의 모바일 콘텐츠를 다른 피쳐폰의 모바일 콘텐츠나 스마트폰 플랫폼인 아이폰의 iOS나 구글의 안드로이드 콘텐츠로, 또는 iOS 앱을 안드로이드 앱이나 안드로이드 앱을 iOS 앱으로 스마트폰 플랫폼 콘텐츠 간의 변환 작업이 단기간 내에 자동으로 진행될 수 있도록 하여 피쳐폰 콘텐츠를 스마트폰 플랫폼으로, 스마트폰 플랫폼 앱을 다른 스마트폰 플랫폼 앱으로 이식하기 위한 개발 과정, 변환 과정에 중복 투자되던 시간 및 기간을 상당히 단축함으로써 생산성을 향상시킬 수 있도록 하였다. 또한, 단축된 시간 및 기간을 신규 콘텐츠나 앱 개발에 투여함으로써 우수한 모바일 콘텐츠의 개발이 가속화되어 모바일 산업의 생산력 증강에 이바지할 것으로 기대된다.

앞으로 콘텐츠의 실행 속도를 높이기 위한 연구와 콘텐츠 변환기 시스템의 기능을 보완, 확장하여 기존의 콘텐츠들을 현재 확산되고 있는 안드로이드, 아이폰, 위도우즈7, 심비안, 바다와 같은 다양한 스마트폰 플랫폼에서 실행될 수 있도록 변환기를 확장할 예정이다.

참고 문헌

- [1] A.V. Aho, M.S. Lam, R. Sethi and J.D. Ullman, *Compilers : Principles, Techniques and Tools*, 2nd Edition, Addison Wesley, 2007.
- [2] C. A. Hsieh & M. T. Conte & T. L. Johnson, "Java Bytecode to Native Code Translation: the Caffeine Prototype and Preliminary Results," Proceedings of the IEEE 29th Annual International Symposium on Microarchitecture, Dec 1996.
- [3] Joshua Engel, *Programming for the Java Virtual Machine*, Addison-Wesley, 1999.
- [4] L. Vinciguerra & L. Wills & N. Kejriwal & P. Martino & R. Vinciguerra, "An Experimentation Framework for Evaluating Disassembly and Decompilation Tools for C++ and Java," 10th Working Conference on Reverse Engineering, pp. 14-23, 2003.
- [5] James Gosling, Bill Joy, Guy Steele and Gilad Bracha, *The Java Language Specification*, Third Edition, Sun Microsystems Inc, Addison Wesley, 2005.
- [6] Sun Microsystems, CLDC Library API Specification 1.0, <http://java.sun.com/javame/reference/apis/jsr030>
- [7] Sun Microsystems, MID Profile 1.0, <http://java.sun.com/javame/reference/apis/jsr037/>.
- [8] J.Steele, Nelson To, *The Android Developer's Cookbook*, Addison-Wesley, 2010.
- [9] Sayed Hashimi, Satya Komatineni, Dave MacLean, *Pro Android 3*, Apress, 2011.
- [10] Goole, Android, <http://code.google.com/intl/>

- ko/android/.
- [11] David Mark, Jack Nutting, Jeff LaMarche, *Beginning iPhone 4 Development : Exploring the iOS SDK*, Apress, 2011.
- [12] Erica Sadun, *The iPhone Developer's Cookbook : Building Applications with the iPhone SDK*, Addison-Wesley, 2009.
- [13] Apple, iOS Reference Library, iOS Technology Overview, <http://developer.apple.com/devcenter/ios>.
- [14] Apple, iOS Reference Library, Cocoa Fundamentals Guide, <http://developer.apple.com/devcenter/ios>.
- [15] Microsoft, Windows Mobile MSDN, <http://msdn.microsoft.com/en-us/library/bb158486%28v=MSDN.10%29.aspx>, 2010.
- [16] YangSun Lee, Jinki Park, SeungWon Na, "Design and Implementation of the Bytecode-to-SIL Translator to Execute Java Programs in Embedded Virtual Machine," International Symposium on Multimedia Applications in Edutainment(MAEDU2005), pp. 34-42, Feb 2005.
- [17] YangSun Lee, SungKyu Choi, "Design and Implementation of the .NET MSIL-to-Java Oolong IL Translator for Embedded Systems," The 5th International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel /Distributed Computing (SNPD'2004), pp. 266-273, Jun 2004.
- [18] YangSun Lee, SeungWon Na, "Java Bytecode-to-.NET MSIL Translator for Construction of Platform Independent Information Systems," LNAI 3215, Vol.3, Springer, pp. 726-732, Sep 2004.
- [19] YangSun Lee, YoungKrun Kim, HyukJu Kwon, "Design and Implementation of the Decompiler for Virtual Machine Code of the C++ Compiler in the Ubiquitous Game Platform," LNCS 4413, Vol.1, Springer, pp. 511-521, Dec 2007.
- [20] 한국 무선인터넷 표준화 포럼, 모바일 표준 플랫폼 규격, <http://www.kwisforum.org/>.
- [21] 박건태, 김승엽, *Jlet으로 배우는 위퍼 프로그래밍*, 한빛미디어, 2005.
- [22] 신지소프트, GNEX SDK, http://www.gnexclub.com/download/download_a1.jsp.
- [23] 신지소프트, Mobile C Library Function Reference, <http://www.gnexclub.com/>.
- [24] 신지소프트, Mobile C Programming Guide, <http://www.gnexclub.com/>.
- [25] 김미영, 무선 인터넷 서비스를 위한 XML 기반 콘텐츠 변환 시스템의 설계 및 구현, 영남대학교 석사학위논문, 2003
- [26] 김영선, 장덕철, "XML Parser 추출에 의한 모바일 콘텐츠 변환 설계," 멀티미디어학회 논문지, Vol.6, No.2, pp. 267-274, April 2003.
- [27] 윤성일, *Mobile 기반의 유무선 플랫폼 통합 변환 시스템*, 한남대학교 박사학위논문, 2003.
- [28] 이영종, *WIPI와 BREW 플랫폼 간 C 언어 기반 솔루션 변환 방법*, 충남대학교 석사학위논문, 2007.
- [29] 박상훈, 권혁주, 김영근, 이양선, "모바일 게임 콘텐츠를 위한 GVM-to-WIPI 자동 변환기의 설계 및 구현," 한국정보처리학회 게임 논문지, 제3권, 제2호, pp. 51-60, December, 2006.
- [30] 박상훈, 권혁주, 김영근, 이양선, "모바일 게임 콘텐츠를 위한 GVM-to-MIDP 자동 변환기의 설계 및 구현," 한국정보처리학회 게임 논문지, Vol.3, No.2, pp. 53-62, Dec 2006.
- [31] 이양선 외4인, "모바일 게임 콘텐츠의 자동 변환을 위한 GVM-to-BREW 번역기 시스템," 한국정보처리학회 게임 논문지, Vol.2, No.1, pp. 49-64, June 2005.
- [32] 이양선, "모바일 콘텐츠의 자동변환을 위한 GNEX C-to-WIPI Java 변환기의 설계 및 구현," 한국멀티미디어학회 논문지, Vol.13, No.4, pp. 609-617, Apr 2010.
- [33] 손윤식, 오세만, 이양선, "소스 레벨 콘텐츠 변환기를 이용한 GNEX C-to-Android Java 변환기의 설계 및 구현," 한국멀티미디어학회 논문지, Vol.13, No.7, pp. 1051-1061, Jul 2010.

- [34] 김재성, 최효준, 이양선, “스마트폰 게임 콘텐츠를 위한 GNEX-to-iPhone 변환기의 설계 및 구현,” 한국멀티미디어학회 논문지, Vol.14, No.5, May 2011.
- [35] 오세만, 이양선, 고평만, “임베디드 시스템을 위한 가상기계의 설계 및 구현,” 한국멀티미디어학회 논문지, Vol.8, No.9, pp. 1282-1291, Sep 2005.
- [36] 이양선, “임베디드 시스템을 위한 가상기계 기술,” 한국멀티미디어학회지, Vol.6, No.2, pp. 36-44, Jun 2002.
- [37] 이양선 외 2인, “Intermediate Language Translator for Execution of Java Programs in .NET Platform,” 한국멀티미디어학회 논문지, Vol.7, No.6, pp. 824-831, June 2004.
- [38] 이양선 외 2인, “JVM 플랫폼에서 .NET 프로그램을 실행하기 위한 MSIL-to-Bytecode 번역기의 설계 및 구현,” 한국멀티미디어학회 논문지, Vol.7, No.7, pp. 976-984, June 2004.
- [39] 이양선, 박진기, “Translation Java Bytecode to EVM SIL Code for Embedded Virtual Machine,” 한국멀티미디어학회 논문지, Vol.8, No.12, pp. 1658-1668, Dec 2005.
- [40] 이양선, 임베디드 시스템을 위한 ANSI C 컴파일러 개발, 서경대학교 산학협력단, 2005.
- [41] 이양선, Retargetable C 컴파일러를 위한 코드 최적화기 개발, 정보통신연구진흥원, 2005.
- [42] 이양선, 임베디드 시스템을 위한 가상기계코드 기반의 C++ 컴파일러 개발, 서경대학교 산학협력단, 2006.
- [43] 오세만, 이양선, 고평만, 임베디드 시스템을 위한 가상기계의 설계 및 구현, 한국과학재단, 2002-2005.
- [44] 이양선, iDTV를 위한 자바 컴파일러와 디버깅 도구 개발, 산업기술평가원, 2006-2008.
- [45] 이양선, 스마트폰 플랫폼을 위한 모바일 콘텐츠 자동 변환기 시스템의 설계 및 구현, 한국연구재단, 2010-2015.



이 양 선

- 1985년 동국대학교 컴퓨터공학과 공학사
- 1987년 동국대학교 대학원 컴퓨터공학과 공학석사
- 1993년 동국대학교 대학원 컴퓨터공학과 공학박사
- 1994년 3월~현재 서경대학교 컴퓨터공학과 교수
- 1996년 3월~2000년 2월 서경 대학교 전자계산소 소장
- 2000년 2월~현재 한국멀티미디어학회 이사
- 2005년 1월~2006년12월 한국멀티미디어학회 총무이사
- 2006년 1월~현재 한국정보처리학회 이사
- 2006년 1월~현재 한국정보처리학회 게임연구회 위원장
- 2009년 1월~2009년 12월 한국멀티미디어학회 부회장, 논문지 편집위원장
- 2011년 3월~현재 (사)스마트개발자협회 이사
- 관심분야 : 프로그래밍언어, 스마트폰 솔루션, 임베디드 SW, 게임평가모델, CT 기술 등