공개SW-09-08

공개SW 기술적 효과성 분석 보고서

공개SW 역량프라자 운영 사업

2009. 12. 15

주관연구기관 **정보통신산업진흥원**

지 식 경 제 부

요약문

공개SW의 개방(Open)과 정보 및 기술의 공유 철학은 단순히 하나의 이슈라기보다는 거대한 트렌드로서 받아들여지고 있다. 2009년 가트너의 보고서는 공개SW가 단순히 하나의 사회적 현상이 아니라 새로운 IT의 시작에 중요한 역할을 지속적으로 수행할 것으로 보고 있으며, 안정성과 신뢰성, 보안성에서 인식의 변화가 일어나고 있다. 동시에 공개SW 기반 표준화가 진행되면서 프로그램 호환성, SW 개발과 이식의 복잡성을 감소시키는 방향으로 진행되고 있다. 또한 웹의 “개방, 공유, 협력, 참여”라는 4가지 문화적인 키워드의 결합은 공개SW의 철학을 더욱 공고히 하고 있다.

소프트웨어 개발환경의 변화는 SW 분야의 새로운 패러다임을 제시하고 있으며, IT 서비스 및 소프트웨어 산업의 융·복합화 현상은 “SaaS(Service as Software)"의 개념을 도출하면서 소프트웨어는 하나의 서비스로서 존재하는 패러다임의 변화를 도출하고 있으며, 이러한 기반은 웹의 진화를 더욱 촉진하고 있다.

공개SW는 더욱 다양하고 복잡한 분야로 발전을 거듭하고 있으며, 공개SW 기반의 커뮤니티 개발 활성화 정책이 전 세계적으로 추진되면서 거대 소프트웨어 업체들의 개방화(Open)가 메인스트림을 형성하고 있다.

공개SW의 개방화는 국내 기술 개발 환경의 변화를 동인하고 있다. 공개SW로 인해 빠르고 유연한 개발을 통한 개발 주기 단축, 공개된 기술에의 접근 용이성으로 인해 국내 기술력의 향상을 이끄는 기술 자립도 향상, 기술습득을 위한 표준화 참여에서 선진 기술 선점 및 동향을 주도하는 표준화 참여의 주체가 되면서 선진 기술 시장에 진입하는 계기를 만들었으며, 공개된 핵심 표준 기술에 대한 정부와

민간의 핵심 기술인력 양성 정책으로 인해 기술 선진국과의 기술 격차를 해소했으며, 특정 벤더 중심의 소프트웨어 시장 구도에서 거대한 글로벌 벤더에 비종속적인 문화가 형성되면서 기술 독과점 방지의 효과를 거두고 있다.

소프트웨어 기술 분야에서 국내 소프트웨어 개발 환경은 비약적인 발전을 거듭하고 있다. 국산 순수 운영체제의 등장, 국산 기업용 소프트웨어의 국내 시장 점유율 증가, 웹 서버 및 웹 브라우저 분야의 기술 향상, DBMS 분야의 국산 소프트웨어 개발, 임베디드 시스템 분야의 기술 향상, 국산 모바일 플랫폼의 개발 등은 국내 소프트웨어 기술 분야의 거대한 성장이라 할만하다.

국내 소프트웨어 기술의 발전과 더불어 “개방”과 “참여”, 그리고 “나눔”의 철학을 갖는 공개SW의 발전을 위한 커뮤니티의 활성화에 대한 정책적인 지원이 필요하다. 안드로이드로 대표되는 구글의 개방화 정책은 전 세계적으로 개발자, 기업, 국가의 참여를 유도하며 개방(Open)에 대한 화두를 던지고 있으며, 기업의 역할을 “공유”와 “참여”의 철학을 지닌 문화로 자리매김하게 하고 있다.

목　　　차

1. 개요 ···· 1
가. 목적 ···· 1
나. 범위 ···· 1
다. 절차 ···· 2

2. 공개SW 기술 동향 ···· 3
가. 공개SW 기술 로드맵 ···· 3
나. 공개SW 표준화 동향 ···· 4
다. 웹의 진화 ···· 6

3. SW 패러다임의 변화 ···· 8
가. 소프트웨어 개발환경의 변화 ···· 8
나. IT서비스 및 패키지 소프트웨어 산업의 변화 ···· 14
다. 새로운 수익 모델 - 상용 공개SW의 등장 ···· 16
라. 기술 개발 기대 효과 ···· 21

4. 소프트웨어 분야별 SW 기술적 효과 분석 ···· 27
가. 소프트웨어 분야별 기술적 효과 분석 개요 ···· 27
나. 운영체제 (OS) ···· 28
다. WEB ···· 36
라. WAS ···· 43
마. DBMS ···· 52
바. IDE ···· 59
사. 임베디드 시스템 ···· 66
아. 모바일 플랫폼 ···· 75

5. 맺음말 ···· 96

참고문헌 ···· 98

1. 개요

가. 목적

□ 공개SW에 대한 기술적 장점 분석 및 연구

□ SW 산업 분야의 패러다임 분석을 통한 기술적 효과성 분석

□ SW 분야별 공개SW의 기술적 효과 분석(OS, WEB, WAS, DBMS 등)

□ 공개SW를 기반으로 소프트웨어 분야별로 국내 개발 현황을 조사

□ 기술적 효과 분석 내용
- ○ 빠르고 유연한 개발
- ○ 기술 자립 기반 확보
- ○ 표준화 주도를 통한 선진 기술 시장 진입
- ○ 핵심 인력 양성을 통한 기술 격차 해소
- ○ 기술 독과점 방지

나. 범위

□ OS, WEB, WAS, DBMS, IDE, Embedded System, Mobile Platform 분야

□ 상용SW에 대응되는 공개SW 분류

[표1-1. 상용SW에 대응되는 공개SW 분류]

분류	소분류	상용SW 제품명	공개SW 제품명
OS	운영체제	MS Windows	Linux
미들웨어	WEB 서버	IIS	Apache
	WEB 브라우저	Internet Explorer	Firefox
	WAS	Weblogic, Websphere	JBoss
	DBMS	Oracle	MySQL
	IDE	MS VisualStudio	Eclipse
임베디드	임베디드	VxWorks, Windows CE	Embedded Linux
모바일	모바일 플랫폼	Symbian	Android, LiMo

□ 국내 공개SW 기반 소프트웨어 분야별 개발 현황 조사

다. 절차

□ 상용 소프트웨어와 공개 소프트웨어의 개요 및 특징

□ SW 패러다임의 변화 분석 및 기술적 효과성 분석

□ 기술 효과성 분석 분야 선정 및 개요/기능 분석

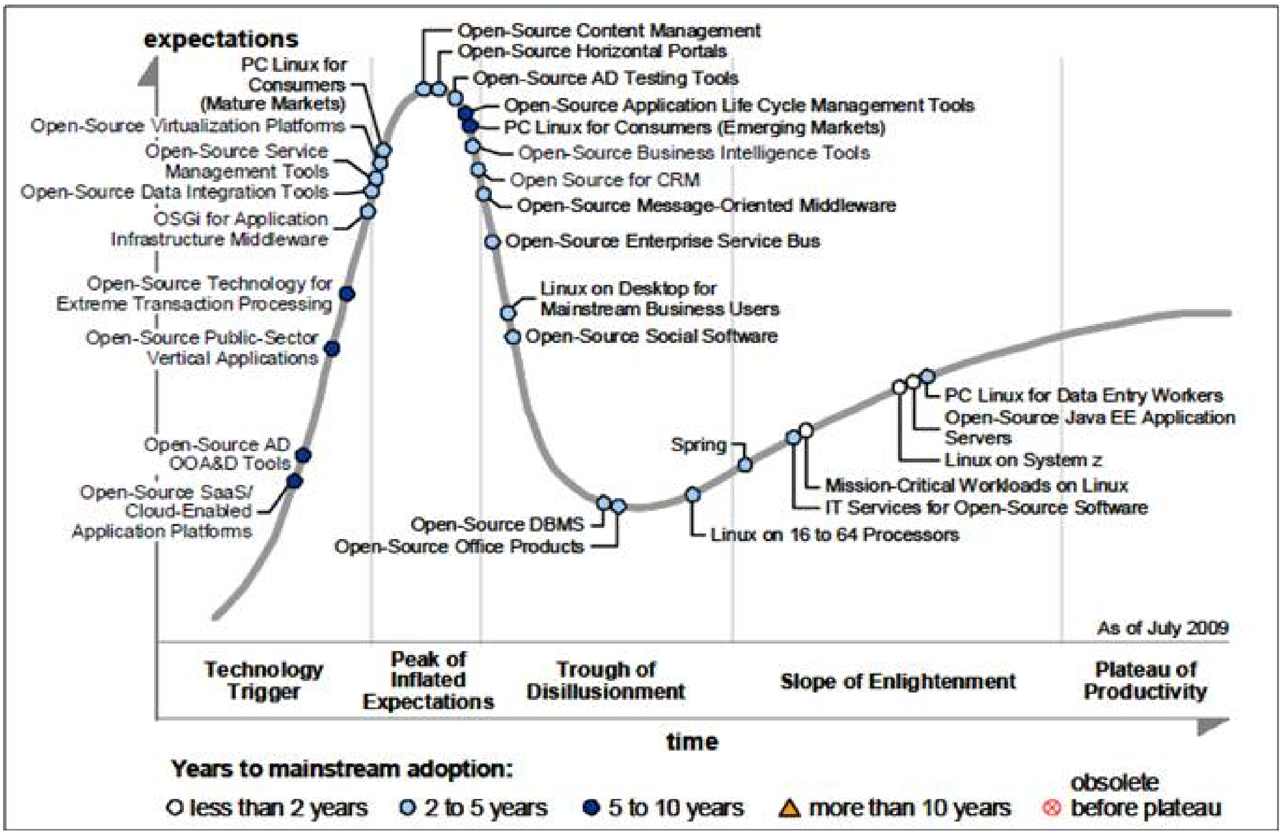
□ SW 분야별 기술적 효과 분석

□ 국내 공개SW 분야별 개발 현황 조사

2. 공개SW 기술 동향

가. 공개SW 기술 로드맵

지식기반 경제에서는 고부가가치를 창출하는 소프트웨어 산업이 국가경쟁력의 핵심 산업이며, 제조업, 서비스업, 정보통신산업 등 주요 기반산업의 경쟁력을 향상시키는 원천이기 때문에 그 중요성이 날로 부각되고 있다. 2009년 가트너의 보고서를 통하여 공개소프트웨어의 발전 방향을 분석할 수 있다.



(Source: Gartner (July 2009))

[그림2-1. Hype Cycle for Open-Source Software, 2009]

보고서는 2년 내에는 Linux와 관련된 기술이 대세를 이룰 것으로 예측했다. 2007~8년의 주된 흐름 즉, 공개소프트웨어는 비공개소프트웨어의 대안으로 점점 비공개소프트웨어를 대체하여 중요시스템으로 그 활용범위를 넓혀가고 있고, 그 채택 이유는 비용 및 위험의 완화라는 2가지는 2009년에도 지속적으로 이어질 것으로 내다봤다. 2009년 공개소프트웨어는 새로운 IT의 시작에 중요한 역할을 지속적으로 수행할 것으로 보고 있으며(예: Social computing, Transaction System), 또한 공개소프트웨어는 Cloud Computing (infrastructure) 분야에서도 중요한 역할을 수행할 것으로 예상 했다. Application Development testing Tools, Enterprise service

bus, Linux on 16 to 64 processors 그리고 DBMS 분야도 많은 발전이 기대되는 분야로 선정했다.

또한 공개 소프트웨어 기술은 개발자와 개발자들 간에 소스를 무료로 공개하며 서로 간의 요구 사항을 반영하면서 기술이 성숙해진다. 즉, 기술 개발이 시작(technology trigger)하여 기술이 시장에 적용되기 전(trough of disillumination)까지는 개발자 커뮤니티를 통해서 기술이 개발된다는 것이다. 이 개발 과정 동안 기업은 기술개발에 필요한 비용을 투입하는 것은 아니지만, 여러 공개 소프트웨어를 가지고 제품화를 하는 단계인 통합, 시험 및 안정화 과정에 소요되는 개발 비용을 투입하여 상용화(plateau of productivity)를 한다. 따라서 공개 소프트웨어기반 제품은 초기 개발 때부터 개발비가 투입된 상용 소프트웨어보다 다소 저렴한 비용으로 구입할 수 있다는 것이지 무료가 아니다.

나. 공개SW 표준화 동향

(1) LSB 규격

1998년 5월경 LSB 프로젝트는 리누스 토발즈, 상용 배포판 업체, 리눅스 인터내셔널, 그리고 Free-BSD의 지원 아래 세상에 나오게 되었다. 현재 LSB 프로젝트는 공개 소프트웨어 기술의 활성화를 지원하는 국제 비영리 컨소시엄인 FSG의 워킹그룹으로 진행되고 있으며, 2001년 6월 최초 규격의 발표를 시작으로 현재 4.0까지 발표하였다. LSB의 목적은 리눅스 배포판 간에 바이너리 응용의 호환성을 증대하고, 하드웨어와 소프트웨어의 이질성으로 발생하는 소프트웨어 개발과 이식의 복잡성을 감소시키기 위해서이다.

LSB에서는 운영체제의 시스템 호출 인터페이스, 시스템 라이브러리, 패키지 포맷 등을 규정하고 있고, 대부분의 리눅스 배포판들의 기준으로 사용되고 있다. 하지만 바이너리 응용의 호환성을 유지한다면, 리눅스 커널이 사용되지 않아도 상관없다. LSB 인증은 FSG의 인증 프로그램을 거쳐야 하며, 제품이 적용되는 인증 시험을 통과했을 때 제품과 관련되는 LSB trademark를 사용하도록 라이선스를 얻게 된다.

(2) FHS 규격

FHS는 소프트웨어로 하여금 설치된 프로그램과 디렉터리의 위치를 예견할 수 있게 하고, 사용자로 하여금 설치된 프로그램과 디렉터리의 위치를 예견 할 수 있게 한다. 1993년 8월부터 시작되어 대부분의 리눅스 배포판은 이 규격을 따르고 있으며, 현재 최신 규격 버전은 2.3이다.

(3) CGL, DCL 규격

CGL, DCL 규격은 OSDL의 산업 요구기능 규격이다. OSDL 기관은 다양한 IT 시스템에 리눅스 기술을 적용시키는 역할을 하는 비영리 기관이다. 2000년에 설립되어 IBM, 인텔, HP, 노키아, ETRI 등 75개 회원사로 구성되어 있다. 리눅스 커널 개발자와 메인테이너인 리누스 토발즈와 앤드류 몰톤 등 45명의 직원으로 구성되어 있다. CGL, DCL 규격뿐만 아니라 OSDL은 DTL, MLI 요구기능 규격도 개발하고 있다. 요구기능 규격 제정은 고객과 벤더의 요구사항과 use case로부터 시작된다. 제안된 요구사항과 use case는 관련되는 내부 워킹그룹에 전달되어, 이미 존재하는 규격과의 중복성, 필요성을 검토한 후 규격으로 추가한다. 다른 규격서와 달리, OSDL은 규격의 구현물을 공개프로젝트로 진행하기 때문에 PoC에 대한 웹 사이트 정보도 함께 제공한다. 따라서 시간이 지나면 OSDL 규격 기능에 대한 구현물은 공개 프로젝트로 완성되며, 벤더들은 그 결과를 제품화로 활용할 수가 있다. 하지만 OSDL은 표준화의 주체는 아니다. 산업체의 요구사항에 따른 문제(gap) 분석, 규격화 및 구현을 한다. 즉 OSDL은 기존의 표준들, LSB, POSIX 등을 준수하고 있다.

CGL 규격은 리눅스를 통신 장비와 같은 실시간성, 고가용성, 고신뢰성을 요구하는 시스템에 적용하기 위한 규격이다. 2002년부터 규격 개발을 시작하여 현재 규격 3.2 버전을 완료하였다. 규격 내용으로는 가용성, 서비스성, 표준, 성능, 하드웨어 지원, 보안, 클러스터링 등 7개 분야이며, 140여 개의 요구 기능 규격으로 구성되어 있다. 2005년부터는 CGL 등록(registration)으로 유통되고 있는 리눅스 배포판의 CGL 규격 만족 정도를 시험하고 있다. 이것은 통신 사업자가 리눅스 제품을 선택하는 기준으로 활용되도록 하고 있다.

DCL 규격은 리눅스를 데이터베이스 시스템과 같은 중요업무 시스템에 활용하기 위한 규격이다. 2002년부터 시작하여 현재 규격 2.1 버전을 완료하였다. 엄밀히 말하면 DCL 경우는 규격이 아니라 capability로 정의하고 있다. DTL은 데스크톱 리눅스 분야이다. 전 세계적으로 3% 정도의 점유율인 리눅스를 더욱더 활성화하고자 한다. 2004년부터 시작되었으나 활동이 미진하였고, 2005년부터 활발한 활동을 하고 있다. MLI은 모바일 시스템에 대한 규격이다. 2005년부터 시작되어 규격 작업이 진행 중에 있다. 다음은 OSDL 활동을 정리한 것이다.

[표2-1. OSDL 활동]

	Carrier Grade Linux	Data Center Linux	Desktop Linux	Mobile Linux
시작년도	2002	2002	2004	2005
목표시장	통신 인프라	엔터프라이즈 서버	데스크톱	단말기, 무선
목표기술	리눅스 커널, 미들웨어	리눅스 커널, 미들웨어	리눅스 커널, 미들웨어, 응용	커널, 미들웨어, 응용, 서비스 전달
회원 수	23	24	19	13
진행상황	요구사항 규격 3.2	Capability 규격 2.1	진행중	진행중

(4) 기타 규격 들

공개 소프트웨어 기술 관련 기타 규격들로는 시스템 자원 관리의 규격인 DMTF의 CIM과 WBEM 표준, 서비스 가용성에 대한 규격인 AIS, 문서포맷의 표준인 ODF 등이 있다.

다. 웹의 진화

(1) 웹 2.0 개념의 진화

2004년과 2005년의 웹 2.0 컨퍼런스를 통해 다양한 기술과 용용, 그리고 성공기업의 사례들이 “웹 2.0”의 이름으로 소개되면서, 본래의 목적과 달리 새로운 기술과 응용을 다루는 “차세대 웹(또는 인터넷) 동향”처럼 지속적으로 확대되었다.

웹 2.0의 개념을 보다 체계화시킨 것은 2005년으로 다양한 논의들이 진행되면서 보다 많은 개념적인 보충들이 진행되었는데, 이때 웹 2.0의 가장 중요한 특징으로 정리된 것이 바로 “플랫폼으로서의 웹”이었다.

기술적인 측면에서는 기존에 진행되어 오던 다양한 XML 응용, SOA, 브라우저 확장기술, RIA, 웹서비스 응용, 시맨틱 웹 응용 등과 같은 차세대 웹 기술과 응용들을 포괄하면서, 여기에 “개방”, “공유”, “협력”, “참여”라는 4가지의 네트워크 기반의 문화적인 키워드들을 결합시키고 있다.

(2) 웹 2.0의 기술적 특징

웹 2.0 자체는 하나의 개별적인 기술 또는 응용으로 출발한 것이 아니라, 복합 비즈니스적 관점에서 출발한 것으로 복합 기술/응용으로써의 기존 기술을 아우르며 차츰 독자적인 개념과 영역을 형성하는 태동단계에 있다. 현재의 태동단계에서 있는 웹 2.0의 핵심적인 기술 요소들과 특징들을 정리한다면 다음과 같이 7가지로 요약할 수 있다.

우선 첫째, 블로그를 이용한 손쉬운 개인 지식의 생성/공유 방식과 함께 표준과 XML에 기반을 둔 유연한 형태의 데이터 교환 방식(RSS, Atom, RDF 등)을 기본으로 한다는 점이다.

둘째, REST, SOAP, WSDL, XML-RPC 등을 이용하는 웹서비스에 기반을 둔 Open API 환경과 개방형 구조를 핵심 요소로 채택하고 있다. 이를 통해 개방적인 시스템 연계 모델과 플랫폼 기반의 서비스 모델로의 발전을 도모하고 있다.

셋째, Open API 등을 통해 데이터 중심의 소프트웨어 통합과 가벼운 프로그래밍 모델을 추구하고 있으며, 인터넷 기반의 서비스 중심의 소프트웨어 환경을 지향하고 있다. 그리고 패키지형태의 소프트웨어 릴리즈 주기가 아닌 지속적으로 업데이트하며 사용자에게 새로운 서비스를 제공하는 소프트웨어 모델로 바뀌고 있다는 점이다.(예: 구글의 서비스 모델들)

넷째, 특히 다양한 업체들과의 Open API를 이용한 협력을 공동으로 추구하며 Open API 기반의 웹서비스들을 사용자들이 조합하여 새로운 서비스를 만드는 매시업(mash-up)과 같은 특징을 가능하도록 한다. (예: 구글맵 + 부동산 서비스, 구글맵 + 택배 추적, 아마존 서비스 + ebay 서비스 + VoIP 등)

다섯째, 보다 편리하고 호환성 있는 사용자 인터페이스 제공을 위해 XML과 웹 표준에 기반을 둔 AJAX, X-Internet, RIA 등의 클라이언트 확장 기술을 발전시키고 있다.(예: 모바일 단말에서의 AJAX)

여섯째, 사용자가 정보를 찾아다니며 브라우징 하는 환경이 아닌, XML로 만들어진 RSS/Atom 등의 정형화된 데이터를 자동으로 구독하여 다량의 정보를 습득하고, 이를 다시 재생산하여 활용할 수 있는 순환 구조로 발전시키고 있다.

마지막으로 일곱째, 다양한 디바이스들과 네트워크 환경을 통합적으로 지원하기 위한 기술들에 대한 연구와 표준화 노력이 진행되고 있다는 점이다. 이를 통해 웹 표준에 기반을 둔 모바일 환경과 데스크톱 환경에서의 응용 호환성을 확보하고, 단말 특성과 네트워크 특성에 영향을 받지 않는 공통의 서비스 플랫폼으로서의 웹 기술을 만들고자 하고 있다.

결국 웹 2.0이란 차세대 웹이 지향하고 있는 다양한 디바이스, 네트워크, 서비스, 데이터 등을 통합하며 보다 편리한 사용자 환경을 제공하기 위한 “플랫폼으로서의 웹” 기술이라고 정의할 수 있다. 이 과정에서 다양한 사용자의 참여를 촉진하고, 그러한 참여에 의해 지식과 서비스의 재생산과 재활용이 촉진될 수 있는 생태계(ecosystem)를 제공한다는 점이 웹 2.0의 가장 중요한 특징인 것이다.

3. SW 패러다임의 변화

가. 소프트웨어 개발환경의 변화

(1) 소프트웨어 공학의 변화

(가) 소프트웨어 공학의 전망

미래사회를 한마디로 표현한다면 유비쿼터스 컴퓨팅(Ubiquitous-Computing) 환경이 지배하는 사회라고 할 수 있다. 유비쿼터스 환경은 다양한 종류의 망, 센스 및 칩, 인간 즉 환경계와 인간계가 끝임 없이(Seamless) 연결되어 활용되는 환경이다. 이 환경에서는 언제 어디에 있어도 인식하지 않고 스트레스 없이 사용되는 신개념 통합 환경(Digital Convergence)이며, 유비쿼터스의 본질은 컴퓨터 및 네트워크가 인간 생활공간의 "상황(Situation)"을 인식하는 것이라 할 수 있다.

이러한 유비쿼터스 환경의 일반적인 특징을 살펴보면 모든 컴퓨팅 장치는 네트워크로 연결, 즉 네트워크에 연결되지 않은 컴퓨터는 유비쿼터스 컴퓨팅이 아니라고 할 수 있다. 또한 컴퓨터는 사용자의 눈에 보이지 않아야(Invisible) 한다. 가상공간이 아닌 현실세계의 어디서나 컴퓨터의 사용(Embodied Virtuality)이 가능하다. 인간화된 인터페이스(Calm Technology)로서 사용자 상황(장소, ID, 장치, 시간, 온도, 명암, 날씨 등)에 따라 서비스가 변하는 특징을 갖는다. 유비쿼터스 컴퓨팅의 특성에 따라 환경을 구성하기 위해 무엇보다도 중요한 것은 모든 장치 및 사물에 컴퓨팅 장치가 내장(Embedded) 된다는 것을 의미한다. 여기에서 유비쿼터스 환경은 임베디드 시스템과 밀접한 관계를 가지며 임베디드 시스템에서의 소프트웨어가 차지하는 비중과 그 중요성이 증대되어 가고 있는 추세이다.

미래사회에서의 소프트웨어의 중요성은 비단 정보기술(IT: Information Technology) 분야의 학자나 일부 IT 기업에서 주장하는 편협한 사고는 아니다. 사회학적인 측면에서는 미래의 사회를 후기 정보화 사회 (PIS: Post Information Society)라고 말하기도 한다. 사회학자 김문조 박사는 미래사회를 고도의 자유도를 지닌 초 개방적 복잡계(Supra-Complexity system of High Degree of Freedom)가 될 것이며 이것이 한국 사회의 표본이 될 것이라고 말한다. PIS는 기술적 고도화 및 융합화, 기술사회적 구성체의 공고화, 사이버 문화를 주축으로 한 제 2의 정보혁명, 이성-감성의 조화에 기반을 둔 고 개념 사회(High Concept Society)라는 특징을 나타내며 이러한 상황에서는 소프트웨어가 사회발전의 요체로 대두될 것이라고 말하였다. 이렇듯 미래사회에서의 소프트웨어의 중요성이 강조되고 있는 가운데, 유비쿼터스(Ubiquitous) 사회는 이러한 소프트웨어의 중요성이 기술적으로 실현되는 환경이라고 할 수 있다.

(나) 소프트웨어의 전망

미래의 환경에 적합한 소프트웨어를 개발하기 위한 소프트웨어 공학 기술을 도출하고 발전방향을 수립하기 위해 미래 사회에서 핵심적인 역할을 할 소프트웨어에 대한 전망을 살펴보고자 한다.

임베디드 유비쿼터스 환경으로 대별되는 미래 사회에서의 소프트웨어의 전망에 대하여 많은 전문가들이 "전체 시스템에서 소프트웨어의 비중이 증대될 것이다" 등 다양한 견해를 제시한다. 다양한 시각으로 미래의 소프트웨어에 대하여 조망할 수 있지만, 본 글에서는 소프트웨어공학 기술적인 측면에서 각 분야별 기술을 도출하기 위한 목적과 관점에서 미래 소프트웨어의 특징을 전망하였다.

"대부분의 시스템에서의 소프트웨어의 비중이 현저하게 증가 할 것이다." 항공기의 경우 현재 80% 이상의 기능을 소프트웨어가 담당하고 있으며, 자동차의 경우도 40%에 이르는 기능을 소프트웨어에 의지하고 있다. 이러한 추세는 정밀한 시스템일수록 더욱 더 소프트웨어의 비중이 증대되고 있음을 알 수 있으며 이러한 추세는 미래에 갈수로 더욱 더 심화될 것이다.

"소프트웨어를 포함한 시스템은 고객과 사용자의 요구 만족을 넘어 가치창출이 요구되어지는 추세이다." 소프트웨어와 경영(Business)과의 밀접한 관계가 심화(Tightly Coupled) 되면서 소프트웨어 개발 시에 제품의 품질과 개발의 성공뿐만 아니라 개발된 소프트웨어가 고객의 가치 창출에 어떠한 도움을 줄 수 있는지 등에 대한 분석과 이의 실현을 위한 노력이 병행되어야 한다.

"미래의 소프트웨어는 다양한 인터페이스(Interface), 상황(Situation)의 변화, 요구(Requirements)의 역동적인(dynamic) 변화 등을 만족시켜야 한다." 임베디드 유비쿼터스 컴퓨팅 환경에 대응할 자가 적응 및 관리 등의 기능이 일반화 될 것이다.

"고품질의 소프트웨어에 대한 인증의 요구가 증가되며, 유비쿼터스 환경의 복잡성을 모델링할 수 있는 기법의 활용이 요구될 것이다." 임베디드 시스템은 많은 경우가 Mission Critical한 시스템이다. 이 경우 실시간 또는 고품질의 시스템 인증에 대한 요구가 증가될 것이다. 또한 다양한 망과 칩 등이 네트워크로 연결된 유비쿼터스의 특징은 이벤트의 흐름이나 상태 폭발(States Explosion) 등의 문제를 효과적으로 다룰 수 있는 방법이 요구된다.

"소프트웨어의 변경을 빠르게 전달하라는 요구가 증가하고, 소프트웨어의 개발과 진화의 명확한 구분이 없어질 것이다." 경영과 기술의 급속한 변화는 이를 지원하는 소프트웨어의 예상치 못한 변경을 빠르게 전달 할 것을 요구하며 급격한 비즈니스의 환경변화와 빠른 변경의 요구는 소프트웨어를 계속 빠르게 진화 시킨다. 이것은 임베디드 시스템의 짧은 수명주기와 맞물려 더욱 더 심화될 것이다.

"소프트웨어의 요구는 사용자 환경을 개인의 맞춤형 서비스를 요구할 것이다." 따라서

미래의 소프트웨어 기술은 Custom-made에서 Generic으로, COTS 컴포넌트를 합성하는 방법의 개발이 필요할 것이며, 노동집약형(Labor-intensive)에서 자산 집약형(Asset-intensive)으로 변화할 것이다.
"컴포넌트 기반의 소프트웨어에서 서비스 지향 소프트웨어 환경으로 개발의 패러다임이 변화할 것이다."기존의 컴포넌트 기반의 소프트웨어 재사용이 서비스들의 통합 구성을 통해 다양한 소프트웨어를 개발할 수 있는 방법도 유용하게 사용될 것이다.
"임베디드 유비쿼터스 환경의 특성에 따라, 짧은 수명주기, 하드웨어와 소프트웨어의 협력개발, 실행 중(Run-time) 소프트웨어의 변경, 소프트웨어 개발을 지원하기 위한 자동화된 도구의 사용이 증가되고 전체 소프트웨어 개발 프로젝트의 개발에도 자동화 된 효율 적인 방법이 적용이 요구될 것이다." 이상에서 제시한 미래 소프트웨어의 다양한 전망들을 바탕으로 요구공학(Requirements Engineering), 아키텍처 및 설계(Architecture & Design), 정형기법(Formal Specification & Mathematical Foundation of SE), 품질 및 테스트(Quality, Validation, Verification and Accreditation), 진화 및 역공학(Evolution and Reverse Engineering), 소프트웨어 재사용(Software Reuse), 소프트웨어 프로세스(Software Process), 도구 및 환경(Tools and Environments), 소프트웨어 프로젝트 관리(Software Project Managements), 임베디드 유비쿼터스 환경에서의 소프트웨어공학(Software Engineering for Embedded Ubiquitous) 등 총 10개 분야의 소프트웨어공학과 관련된 기술들이 나아가야 할 방향을 제시하고자 한다.

(다) 소프트웨어 공학 로드맵 전망

□ 요구공학(Requirements Engineering)

미래의 시나리오를 뒷받침하기 위해 소요되는 요구공학 관련 핵심기술로는 전통적인 요구공학기법과 함께 가치기반 요구공학기법, 런타임(Run-time)을 지원하는 요구공학 기법 등이 요구된다. 아키텍처 및 설계(Architecture & Design) 분야에서는 소프트웨어의 설계기술은 컴포넌트기반에서 아키텍처기반 개발의 패러다임으로 변화하면서 설계결정과 품질속성 사이의 관계분석기술, 뷰 별 표현 및 다중 뷰 통합을 지원하는 아키텍처기술 언어기술, 아키텍처와 코드 사이의 일치성보장, 동적 환경 적응형 시스템 개발을 지원하는 아키텍처설계, 아키텍처 지식의 자산화와 관련된 기술들이 요구된다.

□ 정형기법(Formal Specification & Mathematical Foundation of SE)

이 분야에서는 정형기법 및 수학을 기반으로 한 소프트웨어공학 기술로는 전통적인 정형기법 분야의 기수인 정형 명세기술, 정형 검증기술 그리고 환경구축기술 등의 기술이 임베디드 유비쿼터스 환경에 맞게 Tailoring 되어 사용될 것이다.

□ 품질 및 테스트(Quality, Validation, Verification and Accreditation)
관련된 기술은 기존의 기술을 중심으로 임베디드 환경의 특징에 맞게 적합한 방법으로 Tailoring 된 테스트 프로세스(Test Process), 테스트 기술(Test Technique), 테스트 기반(Test Infrastructure)기술이 요구된다.

□ 진화 및 역공학(Evolution and Reverse Engineering)
이 분야에서는 시스템 동역학(Systems Dynamics)을 포함한 타 분야의 학문을 적용한기법과 기존의 분석기법 및 프로세스 기술이 미래 환경에서 사용될 것이다. 시간에 따라 소프트웨어 변경을 모델링하는 시스템 동역학(System Dynamics), 프로그램 이해(Program Comprehension), 변경 연구(Change Study), 역공학(Reverse Engineering), 공학 프로세스(Engineering Process) 그리고 검증 등의 기술이 요구된다.

□ 소프트웨어 재사용(Software Reuse)
이 분야에서는 아키텍처 및 서비스 기반의 소프트웨어의 재사용을 위한 자산명세기술, 자산개발기술, 자산관리기술을 포함 하여 다양한 환경에 적응할 수 있는 소프트웨어 적응 기술 합성 기술 등이 소요 된다.

□ 소프트웨어 프로세스(Software Process)
이 분야에서는 정량적인 프로세스관리 기법과 프로세스의 최적화기술 자동화 기술 등이 필요하다.

□ 도구 및 환경(Tools and Environments)
이 분야에서는 개발 프로세스의 통합관리 및 자동화를 위한 도구 및 환경 기술이 요구되며, 특히 요구공학, 아키텍처 지원도구 및 환경, V&V 지원도구 및 환경, 소프트웨어 재사용, 역공학 지원도구 및 환경, 소프트웨어 프로세스 효과적 지원 및 관리 도구, 정형기법 지원도구 그리고 소프트웨어 공학 지원도구들의 통합화와 관련된 기술들이 요구된다.

□ 소프트웨어 프로젝트 관리(Software Project managements)
이 부문에서는 제품 및 프로세스 재원 등을 통합적으로 관리할 수 있는 환경이 구축되어야 하며 요구되는 요소 기술들은 기본 기술들로는 제품개발관리 기술, 프로세스 관리 기술, 자원 관리 기술 그리고 정량적인 관리를 위한 측정 기술 등이다.

□ 임베디드 유비쿼터스 소프트웨어공학(Software Engineering for Embedded & Ubiquitous)

이 기술은 임베디드 유비쿼터스 환경에 요구되는 소프트웨어를 효율적으로 개발하기 위한 기술들로 대부분 기술들 기본의 소프트웨어공학의 범주에 포함된다. 하지만 기존의 기술 분야와는 차별되는 패러다임을 통한 접근이 필요하며 임베디드 유비쿼터스라는 환경에 특수하게 적용되는 핵심기술들은 서비스 충돌해결(Service Conflict Resolution), 정황인지(Situation Awareness), 소프트웨어와 하드웨어 Co-design, Incremental Model Checking, 실행 중 검증(Run-time Verification), 보안공학(Security Engineering), 자가 치료(Self-healing), Agile Process등이 요구되며, 이러한 기술들은 새로운 독립된 연구 분야로 취급 된다.

(2) 소프트웨어 개발 방식의 변화

과거 10년간 인터넷과 오픈소스 소프트웨어와 같은 신기술은 개발자들이 소프트웨어를 개발하는 방법을 근본적으로 변화시켜 왔다. 더욱이 분산된 소프트웨어 개발팀들은 처음부터 전체 어플리케이션을 구축하는 것보다 그들 자신들이 소유권을 가진 코드와 재사용할 수 있는 컴포넌트 소프트웨어를 조합하기 위해 협업하고 있다.

유비쿼터스 시대가 도래하면서, 오늘날 개발자들은 손쉽게 전 세계에서 수많은 컴포넌트들을 그들의 소프트웨어 자원으로 통합하여 사용할 수 있다.
개발자들은 소프트웨어 모듈과 라이브러리 심지어 코드 단편들을 포함하여 광범위하고 다양한 고품질의 재사용할 수 있는 컴포넌트들을 획득할 수 있으며 이러한 오픈소스와 함께 사내에서 개발된 소프트웨어와 아웃소서와 같은 써드파티 에서의 모듈 또한 증가하고 있다.

그러한 광범위한 가용성을 가지고 개발자들은 현재 그들의 프로젝트를 위해 제일 유용성 있는 컴포넌트들을 선택하여 최적의 기능과 성과, 신뢰성을 제공하는 어플리케이션을 개발하기 위해 컴포넌트들을 자신의 지적자산에 통합하는데 집중하고 있다.

컴포넌트 기반 개발 모델은 소프트웨어 산업을 근본적으로 변화시키고 있으며 조직의 상업용 및 사내 사용을 위한 소프트웨어 개발에 있어서 타임라인 가속화, 소프트웨어 품질 개선, 개발비용절감 등을 가능하게 하고 있다. 그러나 이러한 장점에도 불구하고 현재의 복합지적재산권 환경에서 지적자산에 대한 복합성을 적절히 관리하지 않으면 비즈니스 및 기술적 위험을 조직에 가져올 수 있다.

□ 컴포넌트 기반의 개발
소프트웨어 어플리케이션의 설계, 개발, 유지보수는 중요한 개발 인력자원과 시간의

투자가 필요하다. 개발조직들은 이미 개발되고 테스트된 컴포넌트들을 재사용 하는 것이 새로운 어플리케이션의 구축에 매우 효과적이라는 것을 경험했다. 컴포넌트 기반 아키텍처의 빠른 채택과 진화는 개발팀들로 하여금 경제적으로나 효율성 측면에서 컴포넌트 재사용을 촉진하고 있으며 전문적 기술을 취득하거나 독립적으로 개발하는데 필요한 시간소비 보다 이미 개발된 컴포넌트를 구매 혹은 사용하는 방향으로 전환되고 있다. 이러한 접근은 더욱 빠르고 적은 비용 그리고 더욱 효과적인 소프트웨어 개발을 가능하게 한다.

이러한 컴포넌트의 재사용에 있어서는 각각의 컴포넌트를 지적 자산으로 고려하고 그들 사용을 최적화하여 완전성을 보호하는 것이 바람직하다. 이러한 지적자산으로서의 관리는 third-party에 의해 외부에서 개발된 외부 컴포넌트와 내부에서 개발된 내부 컴포넌트에 모두 적용된다.

외부적으로 개발된 상업용 컴포넌트를 활용하려고 하는 개발팀들은 전형적으로 컴포넌트의 소유권을 구매하지 않는다. 대신에 그들은 지정된 특정 방법, 특정 프로젝트에 작업하는 개발자의 수, 컴포넌트를 포함하는 제품 탑재 등과 관련하여 각각의 경우에 특정 로열티를 내는 방식으로 컴포넌트 사용 라이선스를 획득한다.
따라서 비즈니스 관점에서 상업적 컴포넌트 라이선싱의 비용은 이익에 의해 보상받을 수 있음을 보증하는 것이 필요하다. 전통적 개발과 이러한 라이선싱 간의 비용분석에는 특정 프로젝트에 컴포넌트 사용을 제한한다든가 로열티 지불 등과 관련한 라이선스의 준법성을 확실히 하는 노력이 필수적이다.

□ 오픈소스 영향
과거 개발에 대한 강력한 새로운 접근으로서 오픈소스 소프트웨어가 현저히 증가되었고 개발된 소프트웨어를 재사용하기 위한 기회가 증가하고 있다. 오늘날 강력한 인터넷 서치 능력을 가지고 개발자들은 빠르게 재사용할 수 있는 소프트웨어 컴포넌트의 광범위한 후보군에서 활용하기에 유용한 컴포넌트 위치를 빠르게 알 수 있다. 재사용 형태는 번들 된 독립 컴포넌트, 라이브러리 링크 및 통합, 통합된 소스코드 혹은 소스코드 단편들을 포함하여 많은 형태를 취할 수 있다. 또한, 이들 컴포넌트들은 기능성, 품질, 성과, 검증을 개선하기 위해 수정되어 질 수 있다.

상업용 소프트웨어 개발을 목적으로 컴포넌트를 취득할 때 외부적으로 개발된 오픈소스 컴포넌트와 단편들의 소유권은 그들의 저작자에게 남는다. 이들 저작자들의 대부분이 초기 사용료나 로열티 없이 소프트웨어의 상업적 사용을 허락하는 반면 많은 이들은 다음과 같은 제약조건을 첨부한다.

○ 속성
○ 사용에 관련된 리포팅
○ 수정과 개선의 공표
○ 동일 라이선스 적용(Replication)
○ 궁극적으로 오픈소스가 되어져야 한다는 것으로 귀착

□ 개발 환경의 변화
○ 소프트웨어를 개발하는 실제 모든 기업들은 현재 복합 IP환경에서 일하고 있음
○ 내·외부적으로 이용 가능한 소스 코드의 재사용은 피할 수 없음
○ 제작된 소프트웨어는 결코 개발사만의 전유물일 수 없음
○ 용도를 막론하고 모든 소프트웨어는 지적 재산권이 존재하고, 따라서 관련 회사들은 반드시 준수해야 할 라이선스 의무조항이 수반

나. IT서비스 및 패키지 소프트웨어 산업의 변화

(1) 융합(Convergence) 현상 심화

□ 패키지 소프트웨어 : 패키지 통합화와 기존 경계 파괴
- 기술 및 제품 중심의 융합 현상으로 운영 시스템, 보안, 네트워크 영역이 상호 통합되어 시스템의 복잡성, 설치비용을 줄이면서 확장성과 통합성을 높여감

□ 시스템 통합(SI : System Integration) : 패키지 전반의 통합 시장으로 확대 및 서비스화, 비즈니스 컨설팅 분야와 결합
- 소프트웨어 제품이 패키지기반의 시스템 통합으로 확대되고, 기술과 서비스의 융합으로 영역이 확장되어 서비스로 인식되어 사용되는 현상이 나타남
- 시스템 통합은 비즈니스 컨설팅 서비스와 결합이 일반화되는 추세

(2) 비즈니스적 측면

□ 패키지 소프트웨어[1])가 "제품"에서 "서비스"로(Saas: Software as a Service) 패러다임 변화
- 소프트웨어의 공급과 유지보수, 업그레이드가 웹을 통해 이루어지면서 기존의

1) 1950년~1960년대 하드웨어의 일부로 인식되어 무료로 번들 되던 시절을 지나, 상용 소프트웨어가 등장하기 시작한 1970년대 중반을 거쳐 1980년대 중반부터 패키지 소프트웨어 산업이 본격 활성화됨. 이후 1990년대 중반 이후 인터넷의 폭발적인 성장기를 겪으면서 변혁기를 맞이함. 1990년대 후반 패키지 소프트웨어 개발업체들은 소프트웨어 판매 방식이 향후 통신 인프라를 기초로 서비스화될 것이라고 예측하였고 2000년대 들어 소프트웨어 중심에서 서비스로 패러다임 전환이 가속화되고 있음

소프트웨어 시장의 라이선스, 유통, 유지보수 모델의 변화 초래
- 전 세계적 Saas 시장은 연평균 21%의 성장을 통해 2008년까지 소프트웨어의 50%이상이 새로운 계약 조건에 의해 구매될 것으로 전망

□ 온-디맨드 서비스(On-Demand Service)
- 대규모의 IT 인프라 투자나 관리, IT환경변화에 따른 추가적 기술인력 확보, 인프라 관리 등을 고려하지 않고, 원하는 시점에 필요한 만큼의 자원을 사용하고 변환하는 모델로, 공급자 중심에서 사용자 중심으로 시장의 변화가 이루어지고 있음
- 최근 들어 전 세계적으로 온-디맨드 방식으로 제공된 기업용 소프트웨어의 시장 규모가 40억 달러를 넘어섰으며, 54%의 성장률을 기록
- 온-디맨드 제공 방식과 관련된 소프트웨어 시장 규모는 2011년까지 연평균 성장률 30%로 145억 달러를 기록할 것으로 전망

□ 소수 소프트웨어 기업에 의한 시장 집중화
- 신기술과 새로운 비즈니스 모델의 출현으로 대형 소프트웨어 기업들은 기업 인수 합병을 통해 새로운 트렌드를 수용하고 있어 시장의 집중화 심화

(3) 기술적 측면

□ 표준화된 서비스 기반 아키텍처(SOA : Service Oriented Architecture)를 통한 소프트웨어 재사용
- 표준에 따른 소프트웨어 개발을 통해 통합성 향상과 서비스 확산을 주도하고 있음
- 다양한 융합 가속화 추세에서 핵심 기술로 중요도가 높아지고 있음
- 2007년 세계 SOA시장 규모는 1,890억 달러로 전체 IT서비스 시장의 약 30%로 전망

□ 기존 소프트웨어 라이프사이클을 변화시키는 오픈소스 소프트웨어 (OSS: Open Source Software)
- 소프트웨어의 소스코드가 공개되어 있어 다양한 개발자가 자유롭게 접속할 수 있고, 이를 재배포 및 수정할 수 있어 기존 상용 소프트웨어보다 빠른 속도의 개발과 개선이 가능해 급속히 기존 상용 소프트웨어 시장을 대체해가고 있음
- 향후 5년 동안 상용 소프트웨어 총매출의 22%이상을 대체할 것으로 전망

다. 새로운 수익 모델 - 상용 공개SW의 등장

(1) 상용 오픈 소스의 개요

	무상	유상
공개	일반 오픈소스 - 카피 레프트 - Sourceforge.org 등	상용 오픈소스 - 카피라이트 - SugarCRM, Alfresco 등
비공개	드라이버, PC 부속 소프트웨어	유상 소프트웨어 패키지 - Oracle, SAP 등

[그림3-1. 상용 오픈 소스의 분류]

[그림3-1]은 소프트웨어의 개발 방향의 공개(소스 코드 공개)/비공개(소스코드 비공개), 그리고 소프트웨어의 유상/무상제공에 따라 단순 구별한 것이다. [그림2]의 좌측 위에 있는‘일반적 오픈소스’란, 흔히 말하는 오픈소스 소프트웨어를 일컫는 것이다. 소스 코드의 공개에 대한 라이선스 조항은 GPL, MPL, 퍼블릭 도메인(public domain) 등 여러 가지가 있지만, 일반적으로는 지적 재산권을 고유의 사기업에 독점시키는 것이 아니라 개발자나 이용자들이 공유하거나, 임의의 비영리 단체에 속하게 하는 것을 일컫는다. 대부분의 경우는 무상으로 제공하고 있으며, 개별 커뮤니티사이트나 Source Forge 등에서 언제든 다운로드 할 수 있다.

같은 무상 소프트웨어라도 비공개이면서 주변 기기의 드라이버나 PC에 서비스에 부속되는 애플리케이션 등이 있다.

또한, 유상이면서 비공개인 소프트웨어의 카테고리에는 유상 소프트웨어 패키지가 포함된다. 오라클(Oracle), 마이크로소프트, SAP 등 저명한 소프트웨어 벤더의 제품은 대부분 여기에 속한다.

상용 오픈소스는 [그림3-1]에서 색깔 있는 카테고리 부분이다. 오픈소스 소프트웨어가 무상인 데에 비해 상용 오픈소스 소프트웨어는 유상이다. 그리고 대부분의 경우 공개돼 있는 소스 코드의 지적 재산권은 특정 사기업에 속해있다. 이것이 이 카테고리의 특징이다.

(2) 상용 오픈 소스의 사례

소스 코드가 공개돼 있으면서 유상이라는 것은 쉽게 이해되지 않는 라이선스 체계이다. 따라서 구체적인 사례를 통해 이 라이선스 체계의 특징을 설명한다.

유럽에는 상용 오픈소스 소프트웨어를 제공하는 전문 벤더가 있는데, 이 글에서는 SugarCRM과 Alfresco의 소프트웨어를 사례를 통해 설명한다.

□ SugarCRM

SugarCRM는 미국 SugarCRM이 제공하는 상용 오픈 소스인 CRM(고객관계관리) 소프트웨어이다. 이 회사는 무상이면서 다운로드가 자유로운 'Sugar 오픈 소스'라는 기본 제품과 유상이면서 소스 코드가 비공개인 'Sugar Professional', 'Sugar Enterprise' 두 가지 타입의 제품을 제공하고 있다.
후자인 유상 버전은 일반적으로 소스코드를 비공개로 하고 있는데, 구입한 엔드유저에게는 소스 코드가 제공되므로, 완전한 비공개가 아닌 세미 오픈 소스라고 할 수 있을 것이다. 유상 버전은 기업에서 본격적으로 CRM을 사용하는 유저를 타깃으로 하고 있으며 Sugar Open Source에 기업용 기능을 풍부하게 부가시킨 것이다.
Sugar 오픈 소스는 Mozilla Public License에서 파생된 SugarCRM 공식 라이선스에 기반을 두어 일반에 공개되고 있으며, 유상 버전은 SugarCRM 독자적인 라이선스 체계에 의해 제공되고 있다. 이처럼 SugarCRM은 소스 코드의 내용과 라이선스 체계가 전혀 다른 두 가지 계통의 제품을 제공하고 있다.

□ Alfresco

Alfresco는 영국의 Alfresco가 제공하는 상용 오픈 소스의 콘텐츠 관리 소프트웨어이다. SugarCRM과 같이 무상 버전과 유상 버전, 두 가지 라이선스 체계로 소프트웨어를 제공하고 있으며, 각각의 차이는 [표3-1]과 같다.

[표3-1. Alfresco의 라이선스 체계]

	Community Network (무상)	Enterprise/Small Network (유상)
사전검증	×	○
보존&업그레이드 서비스	×	○
구축 어드바이스	×	○
소비자 포털	×	○
트레이닝	×	○
라이선스 비용	×	○

두 종류의 라이선스 사이에 소스코드의 차이는 없지만 부가되는 서비스가 다르다.

(3) 유저와 벤더가 가지는 장점

□ 소프트웨어 벤더에 있어서의 장점

▪ 영업•마케팅 비용의 절감

업무용 소프트웨어 벤더의 수익 구조에서 수익의 약 70%는 판매관리비가 차지하고 있다. 즉 유저가 지불한 비용의 대부분 이 영업·마케팅, 혹은 관리에 사용되고 있으며, 개발에 사용되는 비용은 극히 적다.

유저는 벤더의 영업뿐 아니라 구입한 소프트웨어의 보수와 기능 확장에 비용을 나누어 써야 하는데, 실제로는 조금 다르다. 왜 벤더의 판매관리비가 높은 것일까?
그 원인으로, 유저가 해당 소프트웨어를 여유 있게 조작하고 평가한 후 소프트웨어를 구입하기가 어렵기 때문에 제품의 설명은 모두 소프트웨어 벤더 측에서 하고 있다는 것을 들 수 있다.
즉, 소프트웨어 벤더는 여러 차례 유저를 방문하여 제품을 설명하고, 시연해가면서 최적화를 제시하는 등의 작업을 한다. 유저의 인지도를 높이기 위해 막대한 비용을 들여 제품의 광고·선전을 한다는 점도 들 수 있다. 이런 과정에서 영업•마케팅 비용이 높아지는 것이다.
그러나 상용 오픈 소스로 이러한 문제를 해결할 수 있다. 오픈 소스의 특징은 언제라도 무상으로 소프트웨어를 다운로드할 수 있다는 것이다. 많은 유저들이 관심 있는 소프트웨어를 다운로드 하여 동작시켜 실제로 사용하면서 기능과 품질을 평가할 수 있다.
다시 말해, 소프트웨어 벤더 측은 제품의 설명과 시연에 시간을 들일 필요가 없기 때문에 영업 단계가 대단히 짧아지는 것이다. 따라서 벤더의 영업•마케팅 비용이 극적으로 내려가, 라이선스 비용이 낮게 책정된다.
대신, 상품의 평가를 유저가 스스로 하지 않으면 안 된다. 말하자면 유저에 대한 셀프 서비스 형 평가에 의해 저비용이 실현되고 있는 것이다.

▪ 개발 비용의 극적인 저감

리눅스 배급업체(distribute)인 레드햇(Red Hat)과 노벨(Novell)은 리눅스 커널에 독자적인 기술과 보수 서비스를 부가하여 제공하는 상용 오픈 소스 벤더라고 할 수 있다.
여기서는 리눅스의 개발 커뮤니티의 예를 통해, 개발 비용이 저감되는 구조를 해설한다. 최근 리눅스 서버는 연간 백 수십만 대나 출하될 만큼 유저 수가 많아지고 있다.
개발 커뮤니티에서는 상위 100명이 전체 84%의 코드를 개발하고, 그 중 12명만이 44%의 코드를 개발하고 있다. 이 12명 중의 대부분은 레드햇과 IBM, 노벨 등 유명한 배급업체와 통합사업자에 소속된 개발자이다. 그들은 왜 무급개발자로 리눅스 커널의 개발 작업을 하고 있는 것일까.
리눅스 커널의 개발 커뮤니티를 보면 그들은 확실히 무급개발자이지만, 예를 들면 레드햇社 측에서 보면 어디까지나 레드햇 엔터프라이즈 리눅스를 개발하고 있는 유급 엔지니어로 파악할 수 있다.
레드햇은 전 세계적으로 많은 무급개발자 개발자를 거느리고 있으며 대단히 저비용으로 레드햇 엔터프라이즈 리눅스를 개발하고 있기 때문에, 대부분의 시간을 기초 부분의 리눅스 커널 개발에 할애하고 있다. 또 그들이 공헌함으로써 개발 커뮤니티에 더욱 많은 개발자들이 모여들고, 그것이 개발 비용의 저감으로 이어진다.

이는 IBM사도 마찬가지다. IBM은 상용 오픈 소스 벤더는 아니지만, '리눅스를 활용한 업무 시스템을 구축하는 서비스'등을 유상으로 유저에게 제공하고 있다. 이 유상 서비스를 지지하는 리눅스 커널이 IBM의 하드웨어와 소프트웨어에 맞도록 저비용으로 개발하고 있는 것이다.
이같이 유저에게 부가가치 서비스를 제공하는 벤더가 적극적으로 커뮤니티를 리드하여, 더욱 고도의 소프트웨어를 저비용으로 개발해 가는 것이 상용 오픈 소스의 전형적인 형태이다.

▪ 제품 기획 비용의 저하
기존에는 소프트웨어 벤더가 고용한 사원들만 소프트웨어를 개발했기 때문에 기기 확장의 아이디어는 모두 사원들이 가지고 있었다.
그러나 상용 오픈 소스의 경우는, 소스 코드의 전체 혹은 대부분이 커뮤니티에 공개되고 있기 때문에 커뮤니티 멤버에 의한 무한의 이노베이션도 가지게 된다.
SugarCRM의 경우, 세계적으로 2만 명의 커뮤니티 멤버가 있으며, 그들이 24시간 365일, 아이디어를 커뮤니티에 제공하고 있다. 개중에는 기존 코드의 개선이나, 획기적인 아이디어 등 여러 가지가 있으며 아이디어의 총량은 기존의 개발 스타일에 비해 엄청나게 많다.
이렇게 아이디어의 양과 질이 높아짐에 따라 제품 기획에 소비되는 비용이 대폭 저감되어, 소프트웨어의 품질을 보존할 수 있는 것이다.

▪ 높은 보안성
최근 미국 오리건주가 CRM 제품을 선정할 때 가장 보안성이 높은 제품으로 상용 오픈 소스인 SugarCRM을 선택했다. 예전에는 소스 코드가 오픈되면, 악의를 가진 유저들로 인해 소프트웨어로의 침입이 증가하지 않을까하고 우려했지만 실제로는 그 반대라는 것을 알 수 있었다.
커뮤니티 멤버가 많아지면 취약성을 개선할 수 있는 숙달된 엔지니어도 늘기 때문이다. 즉, 항상 품질이나 보안성을 높이기 위한 개발자의 노력이 높아지기 때문에 소프트웨어는 항상 높은 품질 수준을 유지할 수 있는 것이다.

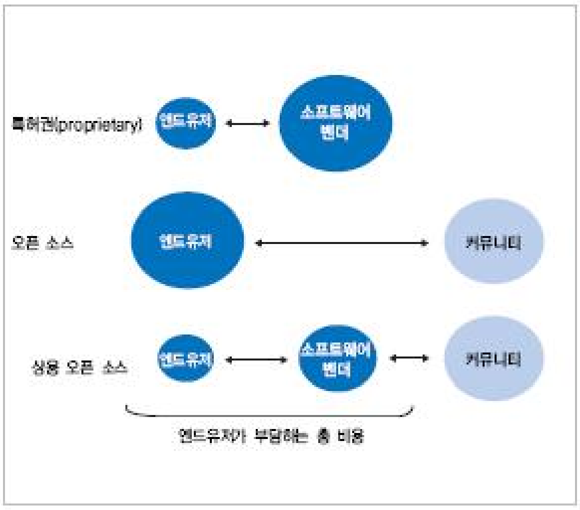
□ 유저에 있어서의 장점
▪ 상용 수준의 기능을 저가격으로 입수
지금까지 설명한 바와 같이 상용 오픈 소스는 오픈 소스에 의한 개발 기법을 도입했기 때문에 소프트웨어 벤더 측의 비용이 저감된다. 결과적으로 유저는 고기능•고품질의 소프트웨어를 저렴하게 입수할 수 있다.

▪ 보수비용의 저하

상용 오픈 소스가 일반 오픈 소스와 가장 다른 점은 소프트웨어 벤더에 의한 전문 보수 서비스가 제공된다는 것이다. 일반적인 오픈 소스의 경우, 책임지고 해당 소프트웨어의 보수 서비스를 제공하는 기업이 없다. 따라서 유저는 커뮤니티에 스스로 참가하여 버그의 해결 방법을 조사하거나 기능 확대를 다운로드하여 검증해야 한다.

또한 요구 사항을 스스로 제안하여 커뮤니티가 받아들이도록 하지 않으면 안 된다. ERP나 CRM 등과 같이 업무에 사용되는 소프트웨어의 경우, 기능 확장이 여러 분야에 걸쳐 대단히 복잡 하므로 소스 코드의 양도 가장 방대하다. 이것을 유저 자신이 커뮤니티와 직접 정보를 주고받으며, 소스 코드를 보수하는 것은 대단히 힘든 일이며, 부담되는 비용(특히, 인건비)도 높아질 것이다. 상용 오픈 소스는 그러한 부분을 소프트웨어 벤더가 지원한다.

그러면 유저가 부담하는 총 비용은 얼마나 될까?



[그림3-2. 유저의 부담 비용]

[그림3-2]는 소프트웨어 제공 스타일에 따라 비용이 어디에 들어가는지 나타내고 있다. 파란 원이 비용의 크기를 나타내고 있으며, 유저는 파란원의 총 비용을 부담하게 된다. 상용 오픈소스에서는 소프트웨어의 보수 서비스를 라이선스 비용에 포함시켜, 전임 엔지니어가 기술적인 지원을 제공하여, 소스코드의 기능을 확장하거나 불량을 점검한다. 또한 커뮤니티가 커질수록 소프트웨어 벤더의 보수비용이 낮아지기 때문에 결과적으로 유저가 부담하는 보수비용도 줄어든다.

라. 기술 개발 효과 및 기대효과

(1) 기술개발 효과

공개SW의 등장으로 가장 큰 변화를 보이는 것은 기술개발 방식이다. 이전의 상업용 소프트웨어는 연구소나 지역단위로 폐쇄적으로 개발을 하였으나, 공개SW는 전 세계적인 네트워크를 통한 개방적인 개발이 이루어지게 된다. 특히 인터넷의 발전으로 인터넷을 통한 집단적 개발 방식이 가속화되고 있다. 이러한 두 가지 유형의 개방방식은 레이몬드의 "성당과 시장"에서 잘 정의되어 있다.

성당식 개발은 기본적으로 상업적 소프트웨어의 개발방식으로 소수의 뛰어난 프로그래머들에 의해 제작되고 장기적인 릴리즈 중심으로 이루어지게 된다(중간판이 없음). 또한 개발과정이나 기능 개선은 주로 개발자들의 판단에 의해 이루어지고 제품의 결함 등에 대한 피드백이 발생하여도 그 대응이 매우 느리거나 전혀 없는 경우도 있다.

시장식 개발의 특징은 프로그래머 자신의 필요에 의해 개발이 시작되고 인터넷에 릴리즈 후 관심 있는 다른 사람들에 의해 개발이 공동으로 진행된다.(병렬적인 테스트와 디버깅이 가능). 반복되는 릴리즈와 피드백을 통하여 사용자 중심의 소프트웨어 개발이 이루어진다. 이러한 시장식 개발이 성공한 가장 큰 이유는 세계에 널리 퍼져 있는 유능한 프로그래머들이 다수 있다는 것이며 인터넷의 발전으로 이들 간의 네트워킹이 결집이 용이해 졌다는 것이다.

시장식 소프트웨어 개발 모델은 첫째, 소프트웨어의 최초모델 작성->공개->사용자의 증가->피드백을 받아 개선->새로운 개발자의 참여->계속적인 발전을 통하여 소프트웨어가 개발되어야 하는데 이때 특히 피드백이 중요하다. 둘째, 특정 OS나 파일 포맷을 사용하지 않고 소스트리를 제공해야 한다. 셋째, 피드백을 위한 의사소통이 중요하므로 원격지 개발자간 신속한 의사소통이 가능하도록 홈페이지, 메일링 리스트, 게시판 등 피드백을 받을 수 있는 공간이 제공되어야만 협동개발의 장점을 살릴 수 있다.

오픈 소스 방식을 통한 소프트웨어 개발 시의 기술 개발 기대효과에 대해서 분석해 보기 전에 먼저 오픈 소프트웨어 중에서 가장 널리 알려져 있는 리눅스의 국내 시장은 현재 어느 상태에 와 있으며 어떠한 문제점이 있는지를 검토해 보면, 먼저 리눅스 활성화의 요인이 되는 주요한 키워드 중심으로 분류해 보면 첫째, 정부의 리눅스 관련 정책이다. 2001년 상반기 정부는 리눅스 활성화를 위하여 리눅스 CD 무료 보급과 관공서 등의 리눅스 채택 유도 등을 추진해 왔다. 그러나 실질적으로 별 효과가 없는 것으로 생각된다. 문제는 취지가 아니라 효용성이다. 리눅스 CD를 보급 시 단지 보급차

원에서 끝나는 것이 아니라 보급을 통해 교육을 시키고, 또한 이를 지속적으로 활용할 수 있도록 방안을 구체적으로 모색하고, 리눅스 CD를 보급하였다면 보급차원이라는 명분에서 벗어나 실질적인 효과를 볼 수 있었지 않았나 생각한다.

둘째, 리눅스 기업의 활동, 리눅스 기업들은 현재 단순한 서버 판매나 호스팅 서비스 등에 의존하고 있다. 그러나 아이러니컬한 것은 리눅스 시장에서 수익모델을 가져가기 위한 전제 조건은 리눅스를 우선 시장에 확산 시켜야 한다는 것이다. 개인시장에서의 윈도우의 아성과 기업시장에서의 유닉스와 윈도우 양분구조에서 리눅스가 자리매김을 하고 리눅스 관련 기업이 성장하기 위해서는 우선은 리눅스를 활성화해야 한다. 아마 이러한 논리를 모르지는 않을 것이다. 그러나 현재 리눅스 업체는 개인플레이를 하고 있다. 대형 리눅스 업체의 자만심과 소형 리눅스 기업의 생계 의존형 사업으로만 치중하고 있는 것이다. 그들이 원하는 것은 빨리 리눅스 시장이 활성화되기를 바라는 마음만 있을 뿐 스스로가 시장을 활성화시키려는 마인드는 없다는 점이다.

셋째, 리눅스 인재양성의 산실인 교육기관의 현주소이다. 200년대 초 리눅스를 전문적으로 배우려는 추세는 엄청난 변화를 가져 왔다. 초기에는 리눅스 관련 교육기관 중 대형교육기관도 거의 교육생을 채우지 못한 상태에서 운영되었지만, 첨차 수강비를 지불해도 강의를 들을 수 없을 정도로 인기를 구가하고 있다. 이로 인해 여러 곳에서 리눅스 교육기관이 우후죽순 생기고 있지만, 대부분이 소수를 제외하고는 전문 교육기관이라기보다는 리눅스 관련 사업을 추진하는 기업에서 하나의 아이템으로 리눅스 교육사업을 하고 있다. 또한 체계화되지 못한 커리큘럼과 전문적이지 못한 강의 방식 그리고 특히 이론 중심의 강의는 리눅스 활성화를 위한 인재양성으로는 너무 약한 교육시스템이다. 이에 따라 리눅스 초급 엔지니어만 양성하고 있고, 실질적으로 필요한 기업형 솔루션들을 개발할 수 있는 개발자 등을 배출시키지 못하고 있는 게 현실이다. 리눅스를 배우는 수강생조차도 리눅스 전문가가 되기보다는 웹마스터를 선호하면서 웹마스터가 되기 위해 윈도우도 배우고 리눅스도 배우는 추세이다.

넷째, 개인이나 기업의 리눅스에 대한 인식, 리눅스를 배우는 개인이 리눅스를 이용 시 가장 중요한 것은 활용성이다. 즉, 엔지니어 등 전문가로 가려고 하지 않는 일반 개인이 리눅스를 배우는 것은 활용성이 있느냐라는 것이다. 리눅스의 설치가 편리해졌다고 해도 리눅스를 통해 활용성이 없다면 설치에 대한 기쁨 이외에는 아무 만족을 못 느끼지 못한다는 점이다. 즉 설치이후 다양하게 활용해 볼 애플리케이션의 절대부족이 개인에게는 리눅스의 활용성을 못 느끼고 있다.
기업측면에서는 리눅스 시스템을 도입 시 엔터프라이즈급 서버로서의 가능성에 대한 의구심과 기술지원 그리고 운영요원 확보에 대한 불안감이다. 그러나 완전히 리눅스의

성능과 잠재성을 인정하지 않는다는 것은 아니다. 즉, 누가 먼저 시도할 것이냐의 문제 일 뿐이다.

한국을 리눅스를 포함한 공개SW 활성화를 통한 소프트웨어 강국을 만들기 위한 대안을 제시하면 다음과 같다.

첫째, 시장을 만들어야 한다. 리눅스 및 이를 포함하는 오픈 소프트웨어는 시장을 만들어야 하는 시점에 있기에 정부, 언론, 기업, 교육기관, 개인 등 모두가 현 시점에서 각자의 이익추구보다는 더욱 큰 시장을 만들기 위한시장 활성화에 주력해야 한다. 국내에서 인터넷이 급속도로 발전할 수 있었던 계기는 초기 인터넷 공개세미나 및 다양한 이벤트의 집중화 그리고 언론의 인터넷 활성화 노력이 일조를 한 것으로 생각된다.

둘째, 리눅스 및 오픈 소프트웨어에 대한 국내 기술기반은 열악하다. 그러므로 현 시점에서는 관련 기업이 상호 경쟁기업이라는 인식을 버리고 시장을 확산하기 위해 관련 벤처 커뮤니티 구축을 이룩해야 한다. 기업 간의 기술교류, 오픈 소프트웨어 수익모델 연구, 사업방향 견학 등 상호 시너지와 기술력을 축적하기 위한 열린 마인드가 중요한 시기라고 본다.

셋째, 리눅스 및 오픈 소프트웨어를 활성화하기 위한 리눅스 공동연구개발센터 설립. 리눅스는 현재 리눅스에서만 가동이 가능하고 사용자가 매력을 느낄 수 있는 킬러애플리케이션을 아주 빠르고 신속하게 시장에 확산시켜야 한다. 그것이 프로그램이거나 서비스 이거나 하는 것은 문제가 안 된다. 현재는 특정 기업이 그러한 역할을 수행하고 있지만, 하나의 기업이 수행하기에는 속도 면에서 매우 비능률적이라고 본다. 그러므로 국내의 관련 기업 모두가 참여하고, 동일한 출자를 통해 관련 기업 모두가 하나의 구성원으로 존재하는 조합형태의 개발 센터를 설립하기를 바란다. 그리고 이러한 개발 센터에서는 국내 시장이 아니라 세계 중심에 우리가 서기 위한 세계 시장을 목적으로 하는 솔루션 공동개발, 개인 시장에서 리눅스와 오픈 소프트웨어의 활용성을 높이기 위해수익성 기반이 아닌 시장 활성화 차원의 킬러애플리케이션 개발, 관련 동호회를 지원하기 위한 지원 센터 운영 등 국내 오픈 소프트웨어의 활성화의 거점의 역할을 하는 센터의 설립은 매우 중요한 시점이라고 본다. 오픈 소프트웨어의 관련 킬러애플리케이션이나 솔루션의 보급 확대의 가장 중요한 관점은 개인이나 기업은 아무리 좋은 솔루션을 무료로 보급한다고 해도 쉽게 적용하려고 하지는 않는다. 그러므로 그러한 서비스를 제공하는 곳에서 이용 가능한 형태로 보급 확산이 시도되어야 하고, 그러한 측면에서 국가가 추진하고 있는 지역정보교육장, 동사무소, 우체국, 학교, 그리고 PC방 등을 통한 무료 배포를 통해 활성화를 더욱더 촉진시켜야 한다.

넷째, 리눅스 및 오픈 소프트웨어가 엔터프라이즈급으로 가기 위해서는 하나의 비즈니스 플랫폼 개념의 레퍼런스 사이트를 구축해야 한다. 현재 대부분의 리눅스 기업이 하고 있는 레퍼런스 사이트는 리눅스의 신뢰성을 주기 위한 모델로는 너무 부족하다. 그러므로 하나의 기업이 비즈니스를 통해 수익을 창출할 수 있는 모든 프로세스 중심에 리눅스를 하나의 통합솔루션으로 구축하는 모델케이스를 만들어야 리눅스가 엔터프라이즈 급 시장으로 발돋움할 수 있으리라 본다.
또한 단지 리눅스만으로 만드는 통합솔루션이 아니라 리눅스를 기반으로 윈도우와 유닉스를 포용하여 구축하는 포용 중심의 솔루션 개발이 절대적으로 필요하며, 상호 멀티 플랫폼에서 최적의 효과를 나타내는 솔루션이 구축된다면 상호모두가 공존할 수 있는 유일한 방법이 아닌가 생각하며, 이러한 통합솔루션 구축도 공공연구개발센터에서 추진하면 가장 효과적일 것으로 보인다.

(2) 기술 개발 기대효과

오픈 소스 공동체는 오픈 소스의 생산성을 증가시키는 방향으로 스스로 조직되어 왔다. 특히 리눅스 세계에서는 개발자와 구별된 별도의 계층을 형성하면서 경쟁하는 여러 개의 배포판 업체가 존재하다는 것이 경제적으로 볼 때 중요한 사실이다.
개발자들은 프로그램을 만든 뒤에 인터넷을 통해서 공개한다. 각각의 배포판 업체들은 이용 가능한 소프트웨어 중에서 원하는 범위에 맞게 소프트웨어를 취합한 뒤에 이를 통합하고 패키지로 만들어 자신의 브랜드를 붙인다. 그리고 고객들에게 판매한다. 고객들은 배포판을 선택할 수 있으며 개발자 사이트로부터 코드를 직접 다운받아서 선택한 배포판을 보충할 수도 있다.

이렇게 개발자와 배포판 업체가 분리된 형태를 통해서 얻을 수 있는 효과는 개선을 위해서 매우 유동적인 내부 시장을 형성할 수 있다는 점이다. 개발자들은 배포판 업체와 일반 사용자들의 관심을 끌기 위해서 소프트웨어의 품질을 놓고 서로 경쟁하며, 배포판 업체들은 그들이 선택한 패키지 선택 정책에 대한 적절함과 소프트웨어에 추가한 부가가치를 통해서 고객들을 유치하려고 경쟁한다.

이러한 내부 시장 구조가 갖는 첫 번째 효과는 각각의 구성단위가 모두 교체 가능하다는 점이다. 개발자는 이탈할 수 있으며, 비록 그들이 만든 코드가 다른 개발자들에게 직접 사용되지 않는다고 해도 다른 사람들의 주목을 받으려고 경쟁하는 경향에 의해서 기능적으로 동일한 대체 코드가 빨리 생겨나게 된다. 배포판 업체가 사업에 실패한다고 해도 공통된 오픈 소스 기반이 손상되거나 영향을 받지 않는다. 오픈 소스에서의 전체적인 사업 환경은 닫힌 소스 운영체제를 기반으로 한 어떠한 거대 기업보다도 시

장의 수요에 대해서 보다 빨리 반응하며, 충격에 대한 내구력과 회복성도 높다.
또 하나의 중요한 효과는 특화를 통해서 고정 비용을 절감하고 효율성을 높일 수 있다는 점이다. 개발자들은 기존의 닫힌 프로젝트에서 고정적으로 일어나는 압력을 경험할 필요가 없다. 무의미하고 짜증스러운 마케팅 대조표를 읽을 필요가 없고 부적절하고 구식이 된 언어나 개발 환경에 대한 경영진의 명령을 들을 필요도 없으며, 제품 차별화나 지적 소유권 보호이라는 명목으로 호환되지 않는 새로운 방법으로 제품을 재개발해야 할 필요도 없다. 또한 가장 중요한 점은 완성일에 대한 마감 기일이 없다는 것이다. 제대로 개발되지도 않은 상태에서 1.0의 버전을 달고 출시되는 일은 없다. 이러한 오픈 소스의 개발 방식은 고품질의 결과를 가져올 뿐 아니라 실제로 결과물을 가장 빨리 생산할 수 있는 방법이기도 하다.

한편, 배포판 업체들은 그들이 가장 효과적으로 만들 수 있는 부분을 특화시킬 수 있다. 따라서 단순히 경쟁력을 유지하기 위해서 수많은 프로그램에 대한 막대한 개발 예산을 낭비할 필요가 없이 시스템 통합이나 패키징, 품질 보증 그리고 서비스 부문에 집중할 수 있다. 배포판 업체와 개발자 모두는 오픈 소스 방식에 있어서 없어서는 안될 요소인 사용자들을 통해서 지속적인 감시와 피드백을 받으며 정직과 투명성을 유지하게 된다.

오픈 소스로의 전환이 완성되면 소프트웨어 세계는 어떤 모습이 될 것인가에 대한 해답을 검토해 보기 위해서 제공되는 서비스가 어느 정도까지 오픈 소스의 기술적 표준에 의해서 표현될 수 있는가라는 기준에 따라 소프트웨어를 분류해 보는 것이 도움이 될 것 같다. 이것은 서비스가 어느 정도까지 공유물화(共有物化, commoditize) 될 수 있는가라는 점과 밀접하게 연관되어 있다. 그러한 기준이 되는 축(軸, axis)은 응용 소프트웨어(모든 것이 공유화되는 것은 아니며 기술적인 공개 표준이 미약하거나 존재하지 않는다.)와 하부구조(모두가 공유할 수 있는 서비스이고 강한 표준이 존재한다.) 그리고 미들웨어(부분적으로 공유되고 완성되지 않은 기술 표준이 존재한다.)를 구분할 때 사람들이 일반적으로 생각하는 기준과 일치한다. 이러한 부문의 대표적인 예는 워드 프로세서(응용 소프트웨어)와 TCP/IP 스택(하부구조) 그리고 데이터베이스 엔진(미들웨어)이다. 만약, 우리가 개별적인 제품이 아닌 소프트웨어 시장 전체에 오픈 소프트웨어를 적용하는 모험을 해 본다면, 다음과 같은 예측이 가능하다.
하부구조(인터넷, 웹, 운영체제 그리고 경쟁 업체의 제품과 호환 되어야 하는 저수준 통신 소프트웨어)는 거의 모두 오픈 소스가 될 것이다. 이러한 부분은 사용자 컨소시엄과 현재의 레드햇사가 하고 있는 역할과 같이, 이윤을 추구하는 배포판이나 서비스 업체들에 의해서 협력적으로 유지될 수 있다. 반면에 응용 소프트웨어는 닫힌 소스로 남아있을 가능성이 매우 높다. 비공개 알고리즘이나 기술의 사용 가치가 소비자들이

돈을 주고 닫힌 소프트웨어를 계속해서 구입할 만큼 충분히 높은 (그리고 비신뢰성에 지불되는 비용이 충분히 낮고 독점 공급에 의한 위험 또한 충분히 감수할 수 있는) 상황이 있을 것이다. 이러한 상황은 네트워크 효과가 미약한 독립된 수직 시장 응용 소프트웨어에서 발생할 가능성이 높다. 이전에 다루었던 제재소의 경우가 한 예이다. 생물학적 인식 소프트웨어도 예가 될 수 있다(지문이나 안구 등의 인체 일부를 이용한 신분 확인 시스템과 같은 분야).데이터베이스와 개발 툴, 커스터마이즈된 최상위 계층의 응용 소프트웨어 프로토콜 스택과 같은 미들웨어는 혼재된 형태가 될 것이다. 미들웨어 분야가 오픈 소스로 향할 지 닫힌 소스로 향할 지는 실패에 대한 위험 비용이 얼마가 되느냐에 따라서 결정될 것이다. 실패에 대한 위험 비용이 높다면 소스를 공개하는데 따른 시장의 압력이 걸린다.

그러나 이러한 구도를 완성시키기 위해서는 '응용 소프트웨어'나 '미들웨어'가 완전하게 고정된 범주는 아니라는 사실을 인식할 필요가 있다. 개별적인 소프트웨어 기술은 자연적인 생명 주기를 갖고 있어서 합리적인 닫힌 상태에서 합리적인 공개 상태로 변해간다. 동일한 논리가 보다 일반적인 상황에도 적용된다.
응용 소프트웨어는 표준화된 기술이 발전되고 서비스의 일부분이 공유화되어 감에 따라서 미들웨어로 변해가는 경향이 있다.(예를 들면, 데이터베이스는 SQL의 등장으로 엔진과 프론트 엔드가 분리되어 미들웨어로 변했다.) 또한 미들웨어 서비스가 공유화되면서 이번에는 오픈 소스의 하부구조로 바뀌는 경향이 있다. - 이러한 변이는 현재 운영체제에서도 직접 일어나고 있다.

미래에는 오픈 소스로부터의 경쟁을 포함한 모든 종류의 소프트웨어 기술들이 결국 사라지거나 공개된 하부구조의 일부분이 될 것으로 예측할 수 있다. 이러한 사실은 닫힌 소프트웨어를 통해서 수익을 얻고 있는 기업자들에게 좋은 소식이 아니겠지만 소프트웨어 산업 전체에는 기업 정신이 계속해서 유지된다는 것을 시사한다.
새로운 시장이 응용 소프트웨어 층과 같은 상위 차원에서 지속적으로 창출될 것이며 제품의 범주가 하부구조로 편입될 때까지 은폐된 지적 소유권의 독점력은 제한된 수명을 갖게 될 것이다.

마지막으로, 두 말할 필요도 없이 이러한 균형은 그 과정을 움직이고 있는 소프트웨어 소비자들에게 좋은 것이다. 제품 생산이 중단되거나 누군가의 통제로 인해 차단되는 대신 보다 많은 고품질의 소프트웨어를 계속해서 이용할 수 있게 될 것이다. 교역이든 증여든 간에 강제되지 않은 모든 활동을 포함한 가장 넓은 의미의 자유론적인 관점에서 자유 시장은 모든 사람들을 위한 소프트웨어의 부를 영속적으로 증가시킬 수 있을 것이다.

4. 소프트웨어 분야별 기술적 효과 분석

가. 소프트웨어 분야별 기술적 효과 분석 개요

(1) 기술 효과성 분석 개요

□ 대표적인 공개SW와 상용SW의 기술적 장점 및 특징 분석
□ 소프트웨어 분야별 기술 동향과 국내 기술 및 개발 동향을 분석
□ 공개SW 기반의 소프트웨어 분야별 국내 제품 개발 현황 조사
□ 기술적 효과성은 다음 사항을 기반으로 국내의 사례를 통해 분석
- ○ 빠르고 유연한 개발
- ○ 기술 자립 기반 확보
- ○ 표준화 주도를 통한 선진 기술 시장 진입
- ○ 핵심 인력 양성을 통한 기술 격차 해소
- ○ 기술 독과점 방지

(2) 소프트웨어 분석 분야

□ OS (운영체제)
□ WEB
□ WAS
□ DBMS
□ IDE (통합 개발환경 도구)
□ Embedded 시스템
□ Mobile Platform

나. 운영체제(Operating System)

(1) 운영체제 기술 동향

최근의 운영체제는 다양한 분야에서 기술적으로 많은 변화가 있으며, 특히 Open Source 정책이 전 세계적으로 확산되고 있는 상황에서 기술과 정보의 공유가 대세를 이루면서 더 급진적으로 변화하고 있다. 하지만 급격한 기술적 발전이 단점으로 작용하기 보다는 장점으로 작용하면서 많은 소프트웨어와 플랫폼이 Open화하는 추세를 주도하고 있다.

몇 가지 주제를 기반으로 운영체제 동향을 분석한다.

□ 그린 운영체제

기존의 서버 컴퓨팅 시스템의 화두는 단연 성능과 안정성이었다. 즉 가격대비 성능과 안정성이 높기만 하면 전력사용량은 별 문제가 되지 않았다. 또한 전력 절감 기술은 전력을 항상 공급받지 못하는 랩톱과 같은 이동형 컴퓨팅장치에만 적용되었다. 하지만 최근의 폭발적인 네트워크의 확산과 인터넷 서비스의 증가에 따라 인터넷 서비스를 제공하는 서버의 숫자와 성능이 급격히 증가함으로 인해 서버를 운영하는 데이터센터(IDC)의 전력사용량이 기하급수적으로 증가하였고, 이에 대한 우려가 심각한 수준이다. 시장조사 업체인 Frost & Sullivan에 따르면, 2007년 총 220만 대였던 미국 데이터센터 내 서버가 2010년경에는 680만 대로 3배 이상 늘어날 것으로 예상한다. 또한 인텔의 보고서에 따르면 2000년 서버 한 대 당 50 와트의 전기를 사용했으나 현재 2009년에는 한 대 당 250 와트의 전기를 소모하고 있다고 한다.
최근의 경향은 모바일에만 적용되던 전력절감 기술을 서버 컴퓨팅 분야에도 적용하여 전체 시스템의 전력사용량을 낮추는 것을 목표로 한다.

□ 가상화 기술

한 시장조사 기관은 2010년까지 물리적 서버 수가 3800만대에 이르고 지난 15년 동안 700% 증가한다고 분석한 바 있다. 이를 수용하는 데 1평방피트당 1000달러가 들고 서버 1대당 평균 2400달러가 소요되며 랙(RACK) 당 평균 4만 달러가 들어간다. 또한 서버에 소요되는 비용의 절반 수준의 비용이 다시 전력과 냉각 비용으로 지출된다. 그런데 서버를 100% 사용하는 것도 아니며 유휴 서버 자원에만 1400억 달러가 소요된다는 보고이다. 이 같은 낭비 요소를 제거하기 위해 가상화 기술을 도입하면 자원 관리 효율성, 가용성, 파일이나 애플리케이션 이동성, 비즈니스 연속성, 인프라 최적화, 전체 애플리케이션 구동 시간 단축 등의 효과를 볼 수 있다.

(2) 상용 OS와 공개 OS 비교

(가) 상용 OS, MS Windows

□ 개요

마이크로소프트 윈도(Microsoft Windows)는 마이크로소프트사가 개발한 컴퓨터 운영 체제이다. 애플 컴퓨터에서 개인용 컴퓨터에 처음으로 도입한 그래픽 사용자 인터페이스(GUI) 운영 체제인 맥 오에스에 대항하여, 당시 널리 쓰이던 MS-DOS에서 멀티태스킹과 GUI 환경을 제공하기 위한 응용 프로그램으로 처음 출시되었다. 현재 전 세계 89% 이상의 개인용 컴퓨터에서 쓰고 있으며, 서버용 운영 체제로도 점차 영역을 확대하고 있다. 일반 사용자들에게 매우 익숙하다는 점과, 호환되는 유명한 응용 프로그램이 많다는 장점을 갖고 있지만, 보안 문제에서는 취약한 부분이 많은 운영 체제로 인식되고 있다. 일부 제품(윈도 NT 초기판)을 제외하고는 모든 제품이 인텔의 x86 아키텍처 기반 프로세서에서만 동작하기 때문에, 이를 통해 두 회사의 개인용 컴퓨터 시장 독점이 더욱 견고해지는 데 대한 비판이 점차 커지고 있다.

□ 윈도우즈 XP

○ 개요

윈도우즈 XP(Windows XP)는 가정용/업무용 컴퓨터, 노트북 컴퓨터, 미디어 센터와 같은 일반 목적의 컴퓨터 시스템에서 사용할 수 있게 마이크로소프트가 개발한 운영 체제 가운데 하나이다. 윈도우즈 XP가 개발 당시에는 휘슬러(whistler)라는 가명으로 불렸다. XP는 eXPerience에서 "XP"를 따서 만들었다.

윈도우즈 XP는 윈도우즈 2000 프로페셔널과 윈도우즈 Me의 뒤를 잇는 제품이며, 윈도우즈 NT 커널을 기반으로 하는 마이크로소프트사의 첫 소비자 지향 운영 체제이다. 2001년 10월 25일에 처음 출시되었으며, IDC에 분석에 따르면 2006년 1월 기준으로 사용자들은 400,000,000본의 윈도우즈 XP 제품을 사용하고 있었다. 윈도우즈 XP의 뒤를 잇는 운영 체제로는 윈도우즈 비스타이며 2006년 11월 8일에 볼륨 라이선스 버전 고객에게 공개되었으며, 2007년 1월 8일에 전 세계 사람들에게 공개되었다.

"홈 에디션"과 "프로페셔널 에디션"으로 나뉘어서 출시되는데, 프로페셔널에는 홈 에디션에서 제외된 고급 보안 및 네트워킹 옵션들이 포함되어 있다. "미디어 센터" 에디션이 2003년에 출시되었는데, 이는 DVD/TV 기능에 초점을 맞추었고, 프로그램 녹화 및 원격 조정을 지원한다. 이 밖에도 터치스크린 입력을 지원하는 "태블릿 PC 에디션"이 출시되었으며, AMD64 프로세서 및 EM64T 확장을 지원하는 인텔 프로세서를 위한 "x64 에디션"이 출시되었다.

윈도우즈 XP는 2008년 6월 30일 공급 종료했고, 2010년에 초저가형 PC를 위한 Home 버전 공급을 중단한다고 발표했다.

○ Windows XP 정보

- 마이크로소프트 윈도우즈 계열의 일부
- 개발자 : 마이크로소프트
- 웹사이트 : 윈도우즈 XP 한국 공식 홈페이지
- 최신 버전 : 5.1.2600.5512 서비스 팩 3 (x86 SP3) (2008년 4월 21일)
- 소스 형태 : 클로즈드 소스, 공유 소스
- 라이선스 : 마이크로소프트 EULA
- 커널 형태 : 하이브리드식 커널
- 업데이트 방식 : 윈도우즈 업데이트
- 지원되는 플랫폼 : IA-32, x86-64,
- 지원 상태 : 2014년 4월 8일까지 연장 지원. 보안 업데이트는 무료 지원. 유상 지원은 계속 사용 가능. 서비스 팩 2는 2010년 7월 13일까지 지원

(나) 공개 OS, Linux

□ Linux 개요

리눅스(Linux)는 컴퓨터 운영 체제이며, 그 커널을 뜻하기도 한다. 리눅스는 자유 소프트웨어와 오픈 소스 개발의 가장 유명한 표본으로 들 수 있다. 리눅스는 다중 사용자, 다중 작업(멀티태스킹), 다중 스레드를 지원하는 네트워크 운영 체제(NOS)이다.

엄밀하게 따지면 이 '리눅스'라는 용어는 리눅스 커널만을 뜻하지만, 리눅스 커널과 GNU 프로젝트의 라이브러리와 도구들이 포함된, 전체 운영 체제(GNU/리눅스라고도 알려진)를 나타내는 말로 흔히 쓰인다. 리눅스 배포판은 핵심 시스템 외에 대다수 소프트웨어를 포함한다. 현재 200여 종류가 넘는 배포판이 존재한다.

초기에 리눅스는 개개인의 애호자들이 광범위하게 개발하였다. 이후 리눅스는 IBM, HP와 같은 거대 IT 기업의 후원을 받으며, 서버 분야에서 유닉스와 마이크로소프트 윈도우즈 운영 체제의 대안으로 자리 잡았다.

리눅스는 데스크톱 컴퓨터를 위한 운영 체제로서도 인기가 늘어가고 있다. 지지자와 분석자들은 이와 같은 성공을 벤더 독립성과 적은 개발비, 보안성과 안전성에서 기인한다고 분석한다.

리눅스는 처음에 인텔 386 마이크로프로세서를 위해 개발되었으나 현재는 다양한 컴퓨터 아키텍처를 지원한다. 리눅스는 개인용 컴퓨터에서부터 슈퍼컴퓨터는 물론 휴대 전화, 개인용 비디오 레코더와 같은 임베디드 시스템까지 광범위하게 이용되고 있다.

□ 이식성

리눅스 커널은 원래 IA-32 Intel 80386 마이크로프로세서를 위해 설계되었다. 그러나 지금은 다양한 컴퓨터 아키텍처를 지원한다. 리눅스는 광범위한 아키텍처에 포팅된 운영체제 중 하나이며, 포팅뿐 아니라 이러한 장치를 위한 배포판 또한 존재한다. 그래서 HP 아이팩과 같은 PDA에서부터 IBM System z9에 이르기까지 다양한 시스템에서 작동한다.

□ 저작권 라이선스 및 등록 상표

리눅스 커널과 대부분 GNU 소프트웨어는 GNU 일반 공중 사용 허가서 라이선스를 따른다. GNU GPL의 조항에 따르면, 수정이 가능한 소스코드를 배포해야 하며 수정된 소스코드는 반드시 같은 라이선스로 배포해야 한다. 때로 이 라이선스는 카피레프트라고 불린다.

1997년에 리누스 토르발스는 리눅스를 GPL로 만든 것이 완벽하다고 말한 적이 있다. 이 라이선스보다 제약이 덜 심한 것도 있다. 예로, 많은 라이브러리가 GNU LGPL(Lesser General Public License)과 같이 제약이 더 줄어든 라이선스를 사용하며 X 윈도우 시스템의 경우에는 MIT 라이선스를 사용한다.

미국의 경우 리눅스는 리누스 토르발스가 등록한 상표이다. 1997년에 일부 개인이 상표를 등록하고 리눅스 배포자에게 상표권 사용료를 내라고 한 적이 있었다. 토르발스 쪽에서도 상표권을 등록하는 데 성공했다. 그는 상표권 침해를 막기 위해서 등록했다고 한다. 대한민국에서도 비슷한 사건이 있었다.

□ 리눅스 커널

리눅스 커널(Linux kernel)은 유닉스 계열 운영 체제의 커널이다. GNU 일반 공중 사용 허가서 버전 2 (GPLv2) 아래에서 공개되었으며 전 세계적으로 배포자들이 개발한 리눅스는 가장 두드러진 자유 소프트웨어 / 오픈 소스의 본보기들 가운데 하나이다.

리눅스 커널은 1991년에 리누스 토르발스에 의해 생긴 말이다. 일찍이 미닉스 커뮤니티가 리눅스 커널에 코드와 개념을 제공하였다. 그 당시 GNU 프로젝트는 자유 소프

트웨어 운영 체제에 필요한 요소를 많이 만들어 냈지만 자체 커널 GNU 허드는 완전하지 않았고 이용성이 없었다. BSD 운영 체제는 법적 문제로부터 헤어 나오지 못했다. 이는 초기 버전의 제한된 기능에도 불구하고 리눅스가 새로운 운영체제를 사용하기 위한 프로젝트로부터 코드를 채용한 개발자들과 사용자들을 빠른 속도로 모았다는 것을 말해 준다. 오늘날 리눅스 커널은 수많은 프로그래머로부터 기여를 받고 있다.

□ 리눅스 커널 상세 정보

- 개발자 : 리누스 토르발스
- OS 계열 :유닉스 계열
- 상태 : 계속 개발 중
- 최신 버전 : 2.6.31.3
- 최신 버전 출시일 : 2009년 9월 24일
- 커널 형태 : 모노리딕 커널
- 라이선스 : GPL
- 웹 사이트 : http://kernel.org

□ 리눅스 배포본

리눅스 배포판은 리눅스 커널, GNU 소프트웨어 및 여러 가지 자유 소프트웨어로 구성된 유닉스 계열 운영체제이다.

회사 차원에서 관리하고 배포하는 레드햇 리눅스, 우분투, 수세 리눅스등도 있고, 커뮤니티 차원에서 관리하고 배포하는 데비안, 젠투 리눅스 등이 있다. 여러 소프트웨어를 모으고 시험하여 배포판을 만든다. 오늘날에는 전 세계적으로 약 300여 가지의 배포판이 존재한다.

○ RedHat

래드햇은 미국 래드햇사의 리눅스 배포본이다. 래드햇의 가장 큰 특징은 그 독특한 패키징 방식인데, RPM(Redhat Package Manager)라고 불리는 이 패키지 방식은 패키지의 추가와 삭제가 간편하게 가능하고 그런 패키지를 만드는 것 자체도 포함하는 패키징 방식이라는 데에서 많은 호응을 얻었으며 래드햇이 가장 널리 사용되는 리눅스 배포본 형식이 되는데 많은 도움을 주었다. 국내에서는 알짜 리눅스가 래드햇 기반의 리눅스 배포본인데, 래드햇의 배포본 릴리즈를 바탕으로 한국어 사용자들이 사용하기 쉽게 여러 가지 한글 패키지와 패치를 포함한다. 최근에는 한국어 설치 프로그램까지 추가되어 더욱 국내 환경에 편리하게 되었다. 래드햇 내의 고등 개발 연구소(Red Hat Advanced Development Labs)에서는 GNOME등의 프로젝트에 대한 지원을 하고 있

으며, 이 회사에는 Rasterman이나 Alan Cox와 같은 유명한 리눅스 프로그래머와 커널 해커들이 있다.

○ Debian
데비안은 리눅스 배포본의 이름이라기보다는 GNU프로젝트의 여러 가지 성과물에 기반하는 운영체제 배포본이다. 이는 커널을 어떤 것에 사용하는지에 따라 데비안 GNU/Linux라 부르기도 하고, 앞으로 GNU의 HURD커널이 발전되어 릴리즈를 낼 수 있는 단계에 이르게 되면 데비안 GNU/HURD라는 새로운 운영체제를 볼 수 있을 것이다. 데비안은 이런 커널 비의존적인 운영체제를 만들기 위해 일하고 있으며, 현재로서는 리눅스 커널을 사용하고 있는 배포본이 주 작업이다. 데비안도 RPM과 유사한 편리함을 위한 패키징 방식인 DEB를 사용하고 있으며, 국내에서는 Debian-kr이라는 데비안 지원 팀이 구성되어 패키지 작업을 하고 있다.

○ Ubuntu
우분투(Ubuntu)는 데스크톱에서 리눅스를 쉽게 사용할 수 있게 만든 리눅스 배포판이다. 데비안 GNU/리눅스(Debian GNU/Linux)를 바탕으로 만들어졌으며 카노니칼의 지원을 받는다. 우분투라는 이름은 남아프리카의 반투어에서 유래된 말로 "네가 있으니 내가 있다"를 뜻한다. 일반적으로 여섯 달마다 새 판이 하나씩 나오는데 이것은 GNOME의 새 판이 나오는 시기와 비슷하다. 데비안 GNU/리눅스와 견주어 볼 때 사용자 편의성에 많은 초점을 맞추고 있다.

○ Fedora
페도라(Fedora, 예전엔 Fedora Core라고 불렸다)는 레드햇에서 후원하는 페도라 프로젝트에서 개발한 RPM 기반의 리눅스 배포판이며, 개인과 기업의 다양한 컴퓨터 환경에서 사용될 수 있도록 만들어져 있으며, 특히 그래픽 기반의 설치 프로그램인 아나콘다와 시스템-환경설정 도구를 이용해 누구나 쉽게 운영체제를 설치할 수 있는 것이 특징이다. 페도라는 그놈과 KDE 환경을 모두 지원하고 있다. 현재 페도라의 주목적은 자유 소프트웨어와 오픈 소프트웨어를 포함하는 것만 아니라, 그러한 기술들의 첨단을 이끌어 나가는 데에 있다.

○ S.u.S.E
S.u.S.E.는 독일과 유럽에서 가장 유명한 리눅스 배포본이지만 그곳에서만 파는 것은 아니며 전 세계를 대상으로 하고 있다. 기본적으로 RPM기반이며, YaST와 같은 자신만의 설정 도구를 포함한다. S.u.S.E.팀의 공헌이라 하면 XFree86 프로젝트에서 지원하지 않는 비디오 칩에 대한 X서버를 작성한 것인데, 이는 이후 XFree86 릴리즈에 포함

되고 있다.

○ CentOS

CentOS(The Community ENTerprise Operating System)는 리눅스계 운영체제 가운데 하나다. 레드햇 엔터프라이즈 리눅스의 소스코드를 그대로 가져와 빌드해 내놓으며 이 과정에서 이루어지는 변형은 레드햇의 상표가 잘리고 그 자리에 CentOS의 상표가 붙는(상표권 분쟁을 피하기 위해) 정도뿐이다. 판수 또한 레드햇 엔터프라이즈 리눅스의 판수를 그대로 가져오며(소수점 아래 숫자는 업데이트 차수를 가리킨다) 오늘날에는 레드햇 엔터프라이즈 리눅스의 행보를 가장 잘 따라가는 운영체제라고 알려져 있다.

(3) 국내 OS 분야 기술 개발 현황

□ Ginux

○ 개요

Ginux2 Server는 기업의 요구 사항을 충족시키는 완벽한 안정성, 성능, 보안 기능을 목표로 기존의 서버 및 응용 프로그램 SW와의 뛰어난 호환성을 갖추고 있으며 기업용 응용 프로그램 및 서비스 구동에 적합한 플랫폼이다.

○ Ginux 특징

▪ 멀티코어 최적화
▪ 장애처리 대응가능
▪ IPSec
▪ Channel Bonding
▪ 32비트와 64비트 프로그램 호환성

□ Asianux

○ 개요

Asianux는 글로벌 벤더들의 강력한 지원을 받는 리눅스 운영체제로 아시아의 표준으로 정착하고 있으며, Asianux는 아시아 각국의 대표 기업이 협력을 통해 그 노하우를 집약하고 협조하면서 아시아 환경에 적합한 리눅스 배포판을 개발하고, 개발된 성과물은 아시아 공통 재산으로, 이를 기반으로 참여 기업은 각국에서 독자적인 비지니스를 전개할 수 있으며, Open Source 이념에 따라 개발 성과를 커뮤니티에 환원하는 Common (공통)·Collaboration (협조)·Contribution (공헌)의 이념을 지향한다.

○ Asianux 특징
▪ 각 국가의 소프트웨어 환경에 최적화
▪ 서로의 우수 분야 기술을 집약
▪ 기술지원, 보안 및 패치 파일 업데이트 부분 협력으로 안정적인 서비스 제공
▪ 각국에서 공통의 인프라 구축
▪ AXCC(Asianux Compatible and Certified) 프로그램으로 다양한 ISV, IHV에 대한 인증
▪ 각 국가 환경에서 다양한 품질테스트를 실시하여 안정성 검증

□ SULinux
○개요
SULinux는 지난 1998년부터 자체적으로 운영해온 "리눅스포털" 사이트를 운영하면서 자체적으로 패키징 했던 배포판을 지난 2005년도에 일반에게 공개하면서 알려지기 시작하였으며 현재 5만 사이트 이상의 서버에서 사용되고 있다.
SULinux 2.0은 (주)수퍼유저코리아의 리눅스 개발연구소인 "한국 리눅스연구소(KLL)"에서 자체개발하고 있는 리눅스 배포판이며 2009년 3월에 TUI 버전을 출시하였고, 5월 GUI 버전을 출시하였다.

○ SULinux 특징
▪ 설치 후 서버용으로 자동 최적화
▪ 서버용으로 최적화된 패키지 구성
▪ 360여개의 관리자 전용명령어에 대한 기본 보안 설정
▪ 서버관리 전용 명령어 내장 (SSU)
▪ 다양한 최신 RAID 컨트롤러 지원
▪ 8TB 이상의 대용량 파일 시스템 제공

(4) 공개SW 기반 운영체제 기술적 효과
□ 빠르고 유연한 개발
공개SW 정책과 세계적인 Open Source 운동은 많은 부분에서 기술과 정보의 공유 및 진보를 이루고 있다. 특히 운영체제의 다양한 핵심 기술들이 공개되면서 국내 운영체제 기술수준은 선진국의 기술 수준을 따라잡고 있으며 대표적인 공개SW인 리눅스 기반의 다양한 배포판과 운영체제의 핵심 기술에 접근 및 개발을 주도하는 개발자들이 등장해서 개발을 주도하고 있다.

□ 기술 자립 기반 확보

독점 업체의 고유 기술 중 하나인 운영체제의 공개 및 기술과 정보의 공유는 국내 운영체제 기술의 기술 자립도를 확립하는 하나의 계기가 되고 있으며, 운영체제의 원천기술은 미약하지만 서버 분야, 모바일 분야, 임베디드 분야 등에서 기술 자립도가 확립되고 있다.

□ 표준화 주도를 통한 선진 기술 시장 진입

운영체제의 핵심 기술인 프로토콜 분야에서 국내 기업 및 기술자들이 표준화를 주도하게 되면서 세계적인 기업 및 기술자들과 공정한 경쟁을 하고 선진 기술을 주도하는 많은 working group이 등장했다.

□ 핵심 인력 양성을 통한 기술 격차 해소

공개된 운영체제 기술을 통해 커널, 디바이스 드라이버, 네트워크 등에 대한 기술적인 교육이 이루어지면서 선진국과의 기술격차가 줄어들고 있다.

□ 기술 독과점 방지

운영체제는 모든 솔루션의 핵심 소프트웨어로서 특정 벤더(MS,)에 의해 국가의 정책이 좌우될 정도의 독점적인 역할을 수행할 수 있는데, 공개SW의 운영체제 등장으로 인해 이러한 독과점이 일정 부분 해소되었다. 하지만 아직도 여전히 독과점은 존재하고 있으며 공개SW의 사용자 확대 및 안정성에 대한 사용자들의 신뢰가 시급하다.

다. WEB

(1) WEB 기술 동향

2009년 역시 웹 지향 방식이 지속적으로 진화하게 될 것이고 이에 따라 다양한 기술들이 선보일 것으로 예상된다. 특히 일반 사용자들을 넘어 기업의 니즈를 충족시킬 수 있는 서비스들이 더욱 확장될 것이다 SaaS(Software as a Service).

2009 IT산업전망 컨퍼런스가 'New IT세상을 선도하는 IT융합 신산업의 미래'라는 주제를 갖고 롯데 호텔 잠실점에서 열렸다. 발표자로 참여했던 IDC코리아 백인형 부사장은 'Worldwide IT Mega Trend'라는 제목의 강연에서 SaaS를 넘어 모든 것의 서비스화(Everything as a Service)에 대한 전망을 제시했다. 웹2.0 데이터 센터의 서비스화와 스토리지의 서비스화, 경영정보의 서비스화 등, 다양한 분야에서 서비스화에 가속도가 생기면서 온라인의 변화를 이끌 것이라고 했다.

구글은 오피스 기능을 가진 소프트웨어를 온라인에서 서비스하면서 소프트웨어에 대한 개념을 바꾸기 시작했다.

S+S 마이크로소프트는 서비스에 중심을 두고 있는 여타의 SaaS 전략과는 달리 소프트웨어와 서비스가 공존하는 방식이 될 것이란 전망을 바탕으로 이 두 가지를 함께 발전시켜나갈 계획이라고 한다. 데스크톱 오피스를 웹으로 확장한 ‘오피스 라이브’와 ‘XBOX 라이브’ 등이 이러한 전략의 예라 할 수 있다. 사용자들이 언제 어떠한 환경에서든 다양한 단말기를 이용하여 웹에 접속, 마이크로소프트 소프트웨어의 풍부한 기능을 이용할 수 있도록 하겠다고 밝히기도 했다.
구글은 모든 클라이언트 기반의 소프트웨어를 인터넷 기반의 서비스가 대체할 것이라고 주장한다. CRM 솔루션을 제공하고 있는 세일즈포스닷컴(Salesforce.com) 역시 소프트웨어가 사라지고 서비스가 모든 것을 대신할 것이라고 주장하기도 했다. 하지만 마이크로소프트는 소프트웨어와 서비스가 경쟁이 아닌 병행하며 발전할 것으로 예상하고 있다. 기업 관점에서 보면, 엔터프라이즈에서 사용하는 ERP, 생산관리 등 LOB(Line Of Business) 애플리케이션은 서비스 형태로 구현되기 어려울 것이란 생각이다. 그 이유는 보안에 대한 우려로 중요한 데이터를 외부에 두는 것을 꺼리며 안정성 또한 중요할 것이기 때문이다. 또한, 모든 업무가 다 서비스로 구현되면 차별화를 통한 경쟁력을 확보할 수 없을 것이고, 핵심 경쟁력이 필요한 부분은 개별 IT 시스템으로 구축하고 일반화할 수 있는 분야는 서비스로 활용하는 것이 현실적인 선택이 될 것이라는 게 마이크로 소프트의 전망이다. 시장은 비즈니스 요구에 부합하는 소프트웨어와 서비스가 모두 필요한 것이다.
구글도 웹 브라우저에 한정된 서비스를 확장하기 위해 위젯이나 툴바를 병행해 제공하고, 세일즈포스닷컴은 모바일 및 오프라인 에디션을 제공하며, eBay는 파워유저에게 경매를 처리하는 데스크톱 에디션을 별도로 제공하여 사용자 경험을 풍부하게 하기 위해 노력한다. 이러한 사례들을 통해서도 소프트웨어와 서비스가 병행할 것이란 예상이 가능하다.

(2) 상용 WEB 서버와 공개 WEB 서버 비교

(가) 상용 WEB 서버, IIS(Internet Information Server)

□ 개요

마이크로소프트 인터넷 정보 서비스(IIS 또는 인터넷 정보 서버)는 마이크로소프트 윈도를 사용하는 서버들을 위한 인터넷 기반 서비스들의 모임이다. 전반적인 웹사이트 용어 안에서, 아파치 웹 서버에 이어 세계에서 두 번째로 가장 잘 알려진 웹 서버이다. 넷크래프트에 따르면, 2007년 10월 기준으로 전 세계 웹사이트의 37.13%와 전 세계 활성화 웹사이트들의 38.23%가 인터넷 정보 서비스를 사용하고 있다고 한다. 서버는 현재 FTP, SMTP, NNTP, HTTP/HTTPS를 포함하고 있다.

□ 특징

- NCSA(National Center for Supercomputing Applications) 유형 매핑 파일
- HTTP(Hypertext Transfer Protocol) 바이트 범위 지원
- 향상된 오류 로깅 형식을 기본적으로 지원
- SQL(Structured Query Language) 데이터를 HTML(Hypertext Markup Language) 형식으로 변환
- IDC(Internet Database Connector) 파일의 서버 변수
- CGI 응용 프로그램이 모든 환경 변수를 받음
- IDC(Internet Database Connector)를 사용한 다중 질의 및 연결 풀링
- Web 브라우저를 통한 서비스 관리
- SSL 키를 만들기 위한 그래픽 도구
- HTML 페이지와 기타 문서의 내용에 색인 붙이기

(나) 공개 WEB 서버, APACHE

□ 개요

아파치 웹 서버(Apache Web Server)는 아파치 소프트웨어 재단에서 관리하는 HTTP 웹 서버이다. BSD, 리눅스등 유닉스 계열 뿐 아니라 마이크로소프트 윈도우즈나 노벨 넷웨어 같은 기종에서도 운용할 수 있다.

아파치 웹 서버는 현재 세계에서 가장 인기 있는 웹 서버이다. 2007년 2월 현재 세계 전체 웹 서버 중 58%를 차지하고 있다.

- 리눅스 운영체제, 아파치 웹 서버, MySQL 데이터베이스, PHP등으로 웹 서버를 운영하는 것을 각각의 머리글자를 따서 LAMP라고도 부르기도 한다.
- Tomcat, Resin 등과 같은 WAS(Web Application Server) 와 같이 사용할 수 있다.
- Open-SSL, Mod-SSL 을 같이 설치하여, Apache의 보안기능을 강화할 수 있다. (http -> https)

□ 특징

▪ 유닉스 쓰레드
  POSIX 쓰레드를 지원하는 유닉스 시스템에서 아파치를 여러 프로세스와 여러 쓰레드로 혼합해서 실행할 수 있다. 많은 경우 확장가능성(scalability)을 높인다.
▪ 새로운 컴파일 시스템
  컴파일 시스템이 autoconf와 libtool을 사용하도록 재 작성되었다. 그래서 아파치 구성 시스템이 다른 패키지들과 좀 더 비슷해졌다.
▪ 여러 프로토콜 지원

아파치는 여러 프로토콜을 서비스할 수 있는 구조를 갖췄다. mod_echo가 그 예로 작성되었다.

- 비유닉스 플랫폼에 대한 더 나은 지원
  Apache 2.0은 BeOS, OS/2, 윈도우즈와 같은 비유닉스 플랫폼에서 더 빠르고 안정화되었다. 이제 아파치는 이들 플랫폼에서 버그가 많고 성능이 느렸던 POSIX 호환층 대신 자체 API로 구현된 플랫폼 특유의 다중처리모듈 (MPM)과 Apache Portable Runtime (APR)을 사용하여 구현된다.
- 새로운 아파치 API
  모듈 API가 2.0에서 상당히 변했다. 1.3의 여러 모듈 순서와 우선순위 문제가 사라졌다. 2.0은 이를 대부분 자동으로 처리하며, 모듈 순서는 이제 더 유연한 훅(hook) 단위로 지정한다. 또, 아파치 서버 핵심 부분을 수정하지 않고 새로운 모듈 기능을 제공하는 함수가 추가되었다.
- IPv6 지원
  하위 Apache Portable Runtime 라이브러리가 IPv6를 지원하는 시스템에서 아파치는 기본적으로 IPv6 소켓을 기다린다. 또,listen,namevirtualhost,virtualhost 지시어가 IPv6 숫자 주소를 지원한다.
- 필터링
  아파치 모듈을 서버로 오고가는 흐름에 대한 필터로 사용할 수 있다. 예를 들어 mod_include의 INCLUDES 필터를 사용하여 CGI 스크립트 출력에서 Server Side Include 지시어를 처리할 수 있다. mod_ext_filter 모듈은 CGI 프로그램을 핸들러로 사용하는 것과 같이 외부 프로그램을 필터로 사용할 수 있게 한다.
- 다국어 오류 응답
  브라우저로 보내는 오류 응답문이 이제 SSI 문서를 사용하여 다국어로 제공된다. 관리자는 통일된 외관을 위해 이 문서를 수정할 수 있다.
- 간단해진 설정
  혼란을 주던 많은 지시어들이 간단해졌다. 자주 혼란을 주던 Port와 BindAddress 지시어는 없어지고 IP 주소 연결에 Listen 지시어만을 사용한다. ServerName 지시어는 리다이렉션과 가상호스트 인식에만 사용될 서버명과 포트를 지정한다.
- Windows NT 유니코드 자체 지원
  Windows NT에서 Apache 2.0은 이제 모든 파일명 인코딩에 utf-8을 사용한다. 파일명은 하위 유니코드 파일시스템으로 직접 변역되어, Windows 2000과 Windows XP를 포함한 모든 Windows NT기반 시스템에 다국어 지원을 제공한다. 이 기능은 Windows 95, 98, ME에는 지원되지 않고, 파일시스템 접근에 전과 같이 시스템의 지역 코드페이지를 사용한다.
- 정규표현식 라이브러리 Updated

Apache 2.0은 Perl호환 정규표현식 라이브러리 (Perl Compatible Regular Expression Library) (PCRE)를 포함한다. 이제 모든 정규표현식에 더 강력한 Perl 5 문법을 사용할 수 있다.

(3) 상용 WEB 브라우저와 공개 WEB 브라우저 비교

(가) 상용SW WEB 브라우저, Internet Explorer

□ 개요

인터넷 익스플로러(Internet Explorer; IE)는 마이크로소프트에서 개발한 웹 브라우저이다. 1995년에 마이크로소프트 윈도 운영 체제에 이 소프트웨어를 기본으로 포함하기 시작한 뒤부터 1999년 이후로 세계에서 가장 널리 쓰이는 웹 브라우저가 되고 있다. 2002년과 2003년에 인터넷 익스플로러 5, 6 버전의 사용률이 95%의 정점에 다다르다가 그 뒤로 인터넷 익스플로러 7 출시에도 불구하고 하락세가 계속되고 있다.

□ 기술적 이슈

그러나 대한민국에서는 인터넷 익스플로러의 의존도가 상대적으로 높다. 이는 온라인 뱅킹 호환성 문제와 더불어 한국의 대다수 웹사이트 개발자들이 여러 웹 브라우저와 운영 체제들의 호환성을 고려하지 않고 현재 마이크로소프트사가 사용을 권장하고 있지 않는 ActiveX를 무리하게 채용하고 있기 때문이다. 또 일부 웹사이트에서는 인터넷 익스플로러 웹 브라우저를 사용하지 않으면 접근할 수 없게 만든 것도 그 까닭의 일부라고 할 수 있다.

□ 시장 점유율

인터넷 익스플로러는 현재 세계에서 가장 많이 사용되는 웹 브라우저이다. 그러나 모질라 파이어폭스, 오페라 등 다른 웹 브라우저들의 점유율이 오름에 따라 2004년 초 90%에 육박하던 인터넷 익스플로러의 점유율은 점차 떨어져서 2007년에는 모든 버전을 합쳐서 60%대를 유지하였다. 그러다가 2009년 9월 현재, IE 6 12.1%, IE 7 15.3%, IE 8 12.2%, FF 3.5 27.3%, FF 3.0 17.2%로 파이어폭스 3.5가 IE를 제쳤다.

(나) 공개SW WEB 브라우저, Firefox

□ 개요

Firefox 3.5는 일년간 개발이 진행된 Gecko 1.9.1 렌더링 플랫폼을 기반으로 하고 있으며 새로운 웹 표준 기술 지원, 성능 및 사용성 향상, 일반 사용자 대상 새로운 기능 추가 등 이전 버전과 비해 많은 발전이 있었다.

□ 특징

- 70개 이상 다국어 지원
- HTML5 <video> 및 <audio> 요소 지원. Ogg Theora 포맷 및 Vorbis 오디오 포맷 지원.
- 개인 정보 데이터 관리 기능 향상. 사생활 보호 모드 지원.
- 더욱 빨라진 웹 사이트 속도 보장. TraceMonkey 자바스크립트 엔진 지원.
- 위치 기반 서비스 이용 편리. 위치 인식 서비스 지원.
- native JSON 지원 및 웹 워커 쓰레드 기능 지원.
- Gecko 레이아웃 엔진 성능 향상으로 더 빠른 웹 페이지 렌더링 지원.
- 차세대 웹 기술 지원 : 웹 폰트, CSS 미디어 지원 및 신규 속성, 자바스크립트 쿼리 선택자, HTML5 오프라인 지원 기능 및 <canvas> 기반 2D 그래픽, ICC 컬러 프로필 및 SVG 변환 기능 제공.

(4) 국내 웹 분야 기술 개발 현황

해외의 주목할 만한 동향 중 하나는 오픈 플랫폼, 오픈 소스, 개방화, 그리고 협력의 확산이다. 여기에는 개방화와 협력 없이는 더 큰 파이를 만들 수 없다는 절박함이 있다. 이처럼 웹 표준을 기반으로 급성장하고 있는 해외의 모바일 웹 2.0 생태계들과 비교하여 국내의 현실은 암울하기까지 하다. 어떤 면에서는 유선 웹 2.0 또는 PC통신의 데자뷰(déjà vu)가 되지 않을까란 우려도 든다.

미약하나마 국내에서도 LGT OZ 서비스를 통해 풀 브라우징과 새로운 모바일 웹 브라우징에 대한 관심이 늘어났고, 한편으로는 풀 브라우징의 한계를 느끼며 현실적 이슈들이 제기되는 계기가 되고 있다. OZ 서비스를 통해 확인할 수 있었던 점들은 첫째, 모바일 웹 활성화를 위해서는 저렴한 정액 요금제의 활성화가 필수적이라는 점, 둘째, 풀 브라우징만으로는 근본적으로 해결할 수 없는 ActiveX와 웹 표준 적용의 문제들이 있다는 점, 셋째, 모바일 웹의 활성화 없이는 차세대 모바일 데이터 서비스의 새로운 시장과 모델을 만들기 어렵다는 점, 넷째, 모바일 웹 생태계의 활성화 없이, 단순히 몇몇 대기업의 계획만으로는 어렵다는 점이라 할 수 있다. 결국 생태계 활성화를 위한 보다 적극적인 이통사의 노력과 웹 표준화의 협력, 유무선의 협력이 필요하다는 점이라 할 수 있다.

모바일 웹 2.0의 동향은 단순한 사용자의 참여, 공유, 개방이라는 의미뿐 아니라, 모바일 웹 환경의 질적 변화와 관련 기술의 질적 변화, 그리고 모바일 산업에서 차지하는 비중의 확산을 포괄하고 있다. 이런 변화의 과정에서 핵심 기술 요소이자 미래의 핵심 플랫폼이 될 모바일 웹 기술, 특히 모바일 웹 브라우징 기술에 대한 다양한 관심과 노력, 그리고 협력이 필요하다.

(5) 공개SW 기반 웹 분야 기술적 효과

□ 빠르고 유연한 개발

국내 공개SW 기반 웹 서버 및 웹 브라우저 기술 부분은 모바일과 임베디드 분야에서 확장이 이루어지고 있으며, 브라우저 핵심 기술보다는 다양한 임베디드 기기와 모바일 단말기에 포팅을 주로 하는 상황이다. 하지만 기술과 정보의 공유 및 공개는 단말기 기반의 웹 브라우저 개발을 독려하고 있는 상황이다.

□ 기술 자립 기반 확보

○ 서버 분야

▪ 아파치 웹 서버를 대체할 만한 공개SW 기반 웹 서버가 존재하지 않음

▪ 아파치 웹 서버의 성능 개선과 모듈 개발에 치중하고 있음

▪ 웹 브라우저 기술 개발 부분은 거의 하지 않음

○ 임베디드 분야

▪ 웹 서버는 다양한 형태로 다양한 임베디드 장비에서 개발이 이루어지고 있음

▪ 웹 브라우저는 미디어 단말기를 기반으로 개발이 이루어지고 있으며 독자적인 기술을 확보하고 있지만 웹 브라우저 엔진 중 하나인 WebKit의 공개를 통하여 더욱 기술적인 발전 속도가 빨라짐

□ 표준화 주도를 통한 선진 기술 시장 진입

웹 서버와 웹 브라우저 표준화 기술을 주도하며 모바일 웹, 시맨틱 웹, 웹 3.0의 표준화에 주도적으로 참여함

□ 핵심 인력 양성을 통한 기술 격차 해소

웹 서버 기반 핵심 인력은 웹 표준화 부분과 웹 서버 구축 및 성능 개선을 기반으로 기술 교육이 이루어지고 있으며 이를 통해 기술 격차가 해소됨

□ 기술 독과점 방지

공개SW 기반 웹 서버인 아파치와 웹 브라우저인 모질라 파이어폭스의 기술 공개 및 정보 공유로 인해 독과점이 방지되고 있지만 대표적인 웹 브라우저인 인터넷 익스플로러의 기술 독과점은 아직 국내에 존재하고 있으며, ActiveX 기반의 웹 브라우저를 운영해야 하는 상황이 있음

라. WAS

(1) WAS 기술 동향

□ WAS가 기존 웹 환경의 한계를 극복할 수 있는 e비즈니스 환경을 위한 기본 플랫폼으로서의 역할을 하고 있다는 인식이 확산

□ 웹서비스 구축 환경을 위한 기반 플랫폼이라는 벤더들의 마케팅 정책과 기존 IT 환경의 통합, 연동의 유연성, 애플리케이션 개발 시간 단축 등을 통한 신속한 ROI 창출과 TCO 절감

□ eBiz 플랫폼으로 꾸준히 성장
한국IDC가 발표한 '한국 WAS 시장 전망 보고서, 2002~2007'에 의하면 2003년 국내 WAS 시장은 9.5%의 성장률을 보여 약 400억 원대의 시장을 형성된 것으로 나타났다. 이는 2002년까지 비교적 고성장을 보여줬던 것에 비해 둔화된 수치이지만 향후 5년간 약 15.6%의 연평균 성장률을 기록하면서 약 770억 규모의 시장이 형성될 것으로 내다봤다.

하지만 업계 관계자들은 "전반적으로 2002년 대부분의 업체가 WAS 도입을 검토했고 지난해 경기 침체 영향으로 WAS 시장의 성장세가 당초 예상보다 저조한 편"이라고 밝혔다. 이에 따라 벤더들은 아직 WAS를 도입하지 않은 중소기업을 타깃으로 시장 확대를 위한 영업과 마케팅에 집중하고 있다.

한국오라클 관계자는 "지난해 경기 침체로 기업의 IT 예산이 삭감되고 투자가 유보되는 가운데 WAS는 상대적으로 지속적인 성장세를 보였다. 금융권과 통신, 제조 분야의 투자가 미진한 가운데 공공 시장이 새로운 수요처로 떠올라 성장세를 유지할 수 있었다"며 "이는 e비즈니스 확대와 맞물려 WAS가 e비즈니스 구현을 위한 기반플랫폼으로 인식됐기 때문"이라고 밝혔다.

이처럼 지난해 예상만큼 시장은 성숙하지 않았지만 향후 WAS 시장은 계속해서 IT 업계의 주요 이슈로 남을 전망이다. 이는 WAS가 기존 웹 환경의 한계를 극복할 수 있는 애플리케이션과 e비즈니스 환경 구축을 위한 플랫폼으로서의 역할을 제공하기 때문이다. 또한 공급 업체들의 웹서비스를 위한 기반 구축을 시작으로 미래 지향적인 마케팅 활동과 적극적인 영업이 효과를 얻고 있기 때문이다.

□ J2EE Vs 닷넷 경쟁 '2라운드'
J2EE는 지난 1997년 처음 스펙이 나온 이후 지금까지 웹에 가장 보편화된 인프라로 자리 잡고 있다. 특히 자바 진영이라는 말이 만들어질 정도로 ISV 및 파트너들이 대거 참가하고 있다.

현재 WAS 제품들의 공통된 특징은 대부분 J2EE 기반으로 개발됐다는 것이다. 각 벤더들은 표준사양으로 EJB(Enterprise JAVA Bean)와 배포 및 실행 환경을 제공하는 컨테이너, 서버 컴포넌트와 연동해 웹 클라이언트에 다이내믹한 화면을 제공하는 서블릿, JSP를 서비스할 수 있는 엔진을 제공함으로써 표준인 J2EE 사양을 만족하는 제품들을 제공해왔다.

특히 지난해 11월에 썬이 J2EE 1.4 스펙을 발표하면서 WAS 벤더들은 표준 스펙에 맞는 업그레이드 제품을 올해 상반기에 출시할 방침이다.

J2EE 1.3 버전 발표 후 J2EE 1.4 버전에는 웹서비스 표준 지원을 강화할 것이라고 밝혔듯이 실제로 1.4 버전에는 웹서비스 표준들이 대거 포함됐다. 웹서비스 표준 규약인 SOAP 메시지에 데이터를 첨부하는 스펙, UDDI 레파지토리의 상호연동을 위한 스펙, SOAP 메시지를 통한 XML 기반의 RPC 기능, UDDI와 ebXML 등록저장소와 상호연동하기 위한 스펙 등이 포함됐다.

이런 상황에서 마이크로소프트의 웹서비스 전략인 닷넷이 등장하면서 닷넷 기반의 개발 툴과 플랫폼들이 새롭게 등장했다. 특히 지난해 5월 출시한 윈도 서버 2003은 한국마이크로소프트의 차세대 서버 운영체제로 일컬어질 정도로 미들웨어 제품이면서 통합과 웹서비스 지원이 강화된 플랫폼으로써 J2EE 진영과 본격적인 경쟁을 시작했다.

이에 따라 WAS 업체들은 닷넷과 J2EE 진영으로 분리된 웹서비스 지원을 위해 편 가르기 식의 경쟁을 치르고 있다. 마이크로소프트를 제외한 WAS 업체들은 업계 표준으로 자리 잡고 있는 J2EE 스펙이 기업 간 연동성이 뛰어나고 웹서비스 플랫폼으로 높은 점수를 주고 있지만 향후 마이크로소프트의 비전인 닷넷이 어떤 방향으로 진행될 것인가 에도 관심을 보이고 있다.

□ BEA·IBM, '시장 강세 여전'
2008년 국내 WAS 시장은 BEA와 IBM의 선두 다툼 속에 오라클과 썬 그리고 국내 업체인 티맥스소프트의 접전 양상을 보여줬다.

BEA코리아는 지난해 개발과 통합이라는 새로운 전략에 따른 '웹로직 플랫폼 8.1'을 발표, 기존 WAS 시장뿐 아니라 통합 플랫폼 시장에도 출사표를 던졌다. 한국IBM은 자사의 유틸리티 컴퓨팅 전략이 'e비즈니스 온 디맨드' 비전에 따라 웹스피어에 대해 파운데이션 및 툴, 비즈니스 통합, 비즈니스 토털에 집중했다. 특히 IBM 익스프레스 소프트웨어 시리즈를 통해 SMB 시장에 집중하는 모습을 보였다.

한국오라클은 그리드 컴퓨팅을 위한 '10g' 전략과 제품을 발표하면서 '오라클 애플리케이션 서버 10g'를 통해 UDDI, SOAP, WSDL 등의 웹서비스 표준을 기반으로 한 개발 툴을 제공하고 있다. 특히 오라클 애플리케이션 서버 10g는 기업이 신속하게 새로운 애플리케이션을 구성하고 기존 애플리케이션을 통합할 수 있는 비즈니스 서비스와 함께 표준 기반의 재사용이 가능한 웹서비스 기술을 강화했다.

티맥스소프트는 웹 애플리케이션 서버인 '제우스(JEUS)'와 자바 개발 툴인 '웹인원 스튜디오(WebInOne Studio)'를 통해 지난해 BEA와 IBM을 근소한 차이까지 추격해 3강 구도 안에 포함됐다고 자평하고 있다.

이밖에 볼랜드코리아, 한국매크로미디어, 한국사이베이스, 한국아이오나테크놀로지, 한국후지쯔 등도 국내 WAS 시장에서 한몫을 했다. 이들 업체들은 WAS시장에서 10% 미만의 점유율을 보이고 있지만 WAS가 애플리케이션 통합과 웹서비스 플랫폼으로 발전하는데 공동의 노력을 보이고 있다.

□ 애플리케이션 플랫폼 스위트로 확장

올해 IT 전 부분에 있어 가장 큰 화두는 기존 IT 투자의 효율적인 재활용과 신규 투자의 ROI에 대한 사전 분석이다. 기존 IT 투자의 효율적인 재활용과 신규 투자에 대한 ROI의 개선을 달성할 수 있는 길은 우선 기업이 보유한 애플리케이션 인프라스트럭처의 개선에서 찾을 수 있다. 이러한 측면에서 경기 회복에 따른 IT 투자의 우선적인 수요는 패키지 애플리케이션이나 하드웨어 증설보다는 애플리케이션 인프라스트럭처와 관련한 부분에서 발생할 것이다.

한국오라클 관계자는 "WAS를 통해 개발자는 애플리케이션 개발비용을 절감할 수 있어 기업들에게 큰 매력으로 인식되고 있으며, 기존에 WAS가 EAI, 기업포털, 웹서비스 등을 지원하는 미들웨어 역할에서 애플리케이션 플랫폼 스위트(APS)로 확장될 것"이라고 밝혔다.

현재 WAS 시장은 EAI, EIP를 아우르는 통합 플랫폼을 적극 활용해 인프라 소프트웨

어로써 입지를 넓혀가고 있다. 또한 WAS가 웹서비스를 위한 주요 플랫폼으로 인식되어짐에 따라 각 기업들에서는 J2EE 기반의 웹서비스를 위한 기반 플랫폼으로서의 WAS를 포지셔닝하고 단품으로서가 아닌 웹 개발을 위한 통합 플랫폼으로 자리 잡아가고 있다. 이에 따라 올해는 기업의 기간계 시스템을 웹으로 구현하는 WAS의 기본적인 기능을 중심으로 통합, 포털 및 개발 솔루션을 하나로 묶어 제공하는 애플리케이션 플랫폼 스위트 형태가 본격화될 전망이다.

□ J2EE의 통합 플랫폼으로의 전환

분산 트랜잭션 애플리케이션의 구축 시, 더욱 많은 개발자들은 안정성과 보안 및 성능을 위해 서버단의 기술을 활용하기를 원하고 있다. 이러한 서버단의 기술 중 가장 대표적인 기술로 J2EE 기술을 뽑을 수 있다. J2EE 플랫폼은 컴포넌트 기반으로 적은 비용으로 빠르게 분산 트랜잭션 애플리케이션을 설계, 개발 및 조립, 운영할 수 있도록 지원하는 컴퓨팅 환경을 제공한다. 또한, J2EE는 이전 기술에 비해 비즈니스 솔루션을 빠르게 시장에 적용할 수 있을 뿐 아니라, 어느 벤더의 API나 제품에 종속되지 않는 플랫폼 독립적인 컴포넌트 기반의 솔루션을 제공해 왔다.

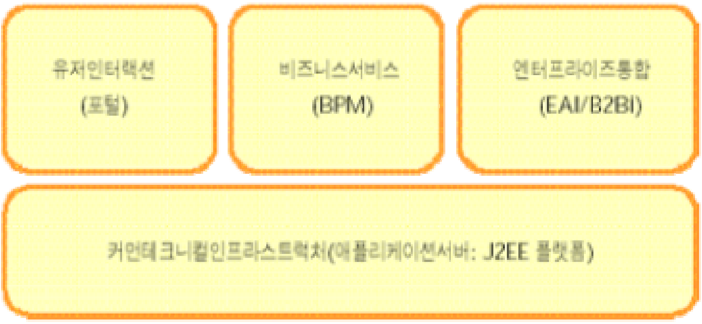
새로이 발표된 J2EE 1.4의 핵심 테마는 웹서비스 지원이다. 이전 버전부터 부분적으로 지원하기 시작한 웹서비스를 1.4에서는 보다 완벽히 지원할 수 있도록 기존 스펙을 보완 및 추가했다. 특히, 웹서비스의 핵심 규약인 SOAP 메시지에 데이터를 첨부하는 스펙(SAAJ), UDDI 레파지토리와의 상호연동을 위한 스펙(JAXR), SOAP 메시지를 통한 XML 기반의 RPC기능(JAX-RPC), UDDI와 ebXML 등록저장소와 상호연동하기 위한 스펙(JAXR) 등이 추가 보완된 스펙들이다. 즉, 자바를 통한 웹 서비스의 구현 및 운영을 보다 손쉽게 할 수 있는 환경을 제공한 것이다.

또, 하나의 테마로는 통합 플랫폼의 기반으로의 확장을 들 수 있다. 즉, CORBA와의 연동강화(RMI Over IIOP), 메시징 미들웨어(JMS), 분산 트랜잭션 처리(JTS, JTA, JDBC) 등의 부분적 보완을 통해 엔터프라이즈 컴퓨팅을 강화했으며, 특히 애플리케이션 통합을 위한 표준 어댑터 기술인 J2EE 커넥터 아키텍처(Connector Architecture) 1.5는 애플리케이션간의 양방향 호출 및 어댑터의 라이프사이클을 추가함으로써 기존의 레거시 및 패키지 애플리케이션의 빠른 통합을 가능케 한다.

□ J2EE 플랫폼 동향

J2EE 플랫폼을 공급하는 많은 벤더들이 올해는 J2EE 1.4에 자신들의 솔루션을 출시할 것으로 예상할 수 있다. 하지만 J2EE 플랫폼 기반의 웹 애플리케이션 서버(WAS)만을 공급하는 것이 아니라 이러한 공통 기술 인프라스트럭처(J2EE 플랫폼)에 기반을 둔 서

비스지향구조(SOA)의 솔루션들로 무장해 더욱 경쟁이 치열할 것으로 예상된다.



[그림4-1. 서비스지향구조(SOA)의 컴포넌트]

□ 벤더 간 WAS 차별성

벤더들이 제공하는 웹 애플리케이션 서버 제품들 간에 혹은 벤더들 간의 차별성은 분명히 존재하고 있다. 그렇다면 과연 그것은 무엇인가?

우선, 웹 애플리케이션 서버의 역할이 기존의 미들웨어적 성격에서 전반적인 기업 애플리케이션 인프라스트럭처의 중심역할을 하는 것으로 확대되면서 중요성이 강조되게 된 것에서 기인한다. 이러한 중요성에 대한 인식의 변화는 웹 애플리케이션 서버의 선정 기준에서 기능적인 요소에 대한 배점보다는 안정적이고 쉽게 확장 가능한 운영 환경을 담보해줄 수 있는 제품의 신뢰성과 확장성에 대한 배점이 보다 더 큰 비중을 차지하게 되는 결과를 이끌어 내고 있다.

선두권 벤더와 후발 벤더와의 차별성과 관련해서 최근에 눈에 띄는 경향 중의 또 다른 하나는 웹 애플리케이션 서버를 중심으로 통합 서버나 포털 프레임워크 경우에 따라서는 이들을 위한 통합 개발 환경까지를 아우르는 통합 애플리케이션 플랫폼 스위트의 시장에서의 약진이다.

지금까지 웹 애플리케이션 시장을 선도해왔던 기업들은 시장에서 제기된 웹 애플리케이션 서버의 확장된 역할에의 요구를 이러한 통합 애플리케이션 플랫폼 스위트를 통해서 만족시킴으로써 벤더 스스로의 사업 영역을 확대해 나감과 동시에 자사 웹 애플리케이션 서버 자체의 경쟁력도 함께 재고해나가고 있다.

(2) 상용 WAS와 공개 WAS 비교

(가) 상용 WAS

1) 웹 로직

□ 개요

BEA 웹로직 서버는 엔터프라이즈급(enterprise level), 다중계층(multi-tier) 그리고 완전한 분산 웹 애플리케이션을 지원하는 자바 애플리케이션 서버다. 웹로직 서버는 시장의 선두주자로서 그리고 사실상 자바 기반 전자상거래 개발 및 운영의 사실상의 표준으로 잘 알려져 있다.

□ 특징

- 웹브라우저, 애플릿 그리고 애플리케이션 클라이언트들을 포함한 다양한 클라이언트들을 관리할 수 있다.
- 신뢰성(reliability), 확장성(scalability) [역주: scalability라는 말은 사실 어떤 서버의 처리 능력이나 업무의 양을 수치화할 수 있다는 의미이다. 그 말은 웹로직과 같은 WAS를 사용할 경우 자신의 시스템이 어느 정도의 부하를 견딜 수 있는지 쉽게 파악이 가능하기 때문에 필요한 시스템 산정 및 관리에 훨씬 유리하다.] 그리고 고성능을 보장하기 위해 웹과 엔터프라이즈 자바빈즈(이하 EJB)의 클러스터링을 지원한다.
- 안전하고 확장 가능한 웹기반의 애플리케이션 개발에 필요한 기반 서비스들을 제공한다.
- Java 2 Platform Enterprise Edition(이하J2EE) 스펙의 최근 버전을 가장 완전하게 지원하고 있다.

BEA 웹로직 서버는 클라이언트나 데이터베이스 커넥션들을 효율적으로 관리하여 시스템 자원을 최대한 활용할 수 있으며 이로 인해 시간당 수십, 수백만 사용자들의 요청을 처리해야 하는 전자상거래 애플리케이션을 지원할 수 있다.

□ 장점

- 컨테이너-컴포넌트 모델
- 컴포넌트 지원과 확장성

2) 웹 스피어

□ 개요

WebSphere는 IBM의 통합 소프트웨어 플랫폼이다. “24x7”, 온 디맨드 웹 애플리케이션과 크로스 플랫폼, 크로스 제품 솔루션을 작성, 실행, 모니터 하는데 필요한 서버, 서비스, 툴 같은 미들웨어 기반 구조들이 포함되어 있다. WebSphere는 믿을 수 있고 유연하며 강력한 통합 소프트웨어를 제공한다.

□ 특징
WebSphere Application Server는 인프라의 기반이다. 모든 것이 이것을 기반으로 구동된다. WebSphere Application Server와 WebSphere Enterprise Service Bus에 기반하고 있는 WebSphere Process Server는 서비스 지향 아키텍처(SOA), 모듈식 애플리케이션의 토대이고, 비즈니스 프로세스를 지원하는 애플리케이션을 구동하는 비즈니스 규칙 애플리케이션을 지원한다. 고성능 환경은 베이스 인프라의 일부로서 WebSphere Extended Deployment를 사용한다. 기타 WebSphere 제품들도 광범위한 서비스들을 제공한다.

WebSphere는 오픈 표준에 기반을 둔 모듈식의 플랫폼이다. 믿을 수 있는 인터페이스를 사용하여 기존 자산들을 WebSphere에 연결할 수 있고, 필요에 따라 환경을 확장시킬 수 있다. WebSphere는 Intel, 리눅스, z/OS 등 많은 플랫폼에서 구동 된다

(나) 공개SW 기반 WAS

1) JBoss

□ 개요
JBoss 어플리케이션 서버 또는 JBoss AS라 불리는 오픈소스의 Java EE 기반의 어플리케이션 서버다. Java 기반의 JBoss 어플리케이션 서버는 Java가 설치만 되어 있다면 어떠한 운영체제이든 지원이 가능한 크로스 플랫폼 기반의 서버다.

□ 특징
JBoss는 100% 자바로 구현된 J2EE 기반의 어플리케이션 서버이며 오픈소스 프로젝트로 진행되고 있다. JBoss는 모든 기능이 JMX(Java Management Extensions) 기반에서 수행된다.
JMX는 다양한 플랫폼과 네트워크 프로토콜에 관련된 문제의 해결책을 제시하는 자바 환경의 장점을 관리 기능에서 사용하기 위해 개발되는 것이다.
JBoss는 Clustering, Failover, Load balancing, Distributed caching, Distributed deployment, Enterprise Java Beans, Aspect-Oriented Programming(AOP) 지원 등 상용의 웹 어플리케이션 서버에서 지원하는 거의 모든 기능을 지원하고 있으며, 각종 포럼이나 개발자 홈페이지 등에서 많은 자료를 얻을 수 있고, Redhat사에서 유상으로 기술 지원을 받을 수도 있다.
크로스 플랫폼 : 소프트웨어나 하드웨어 등이 다른 환경의 운영체제에서 공통으로 사용되는 것. 예를 들면 1개 기종의 하드웨어가 워크스테이션 또는 PC에서 사용되거나, 유닉스 시스템이나 윈도, Mac OS 등의 복수 환경에서 사용되는 것을 말한다.

(3) 국내 기술 개발 현황 및 전망

□ 제우스

티맥스소프트(대표 문진일 www.tmax.co.kr)는 국내 웹애플리케이션서버(이하 WAS) 시장에서 6년 연속 시장점유율 1위를 차지했다고 20일 발표했다. 이번 티맥스소프트의 발표는 한국IDC에서 최근 발표한 '한국 애플리케이션 서버 소프트웨어 플랫폼(ASSP) 2009-2013 전망과 분석, 2008년 리뷰' 보고서를 인용한 것이다.

한국IDC가 발표한 2008년 국내 애플리케이션 서버 소프트웨어 플랫폼(ASSP) 시장 자료에 따르면 티맥스소프트는 전체 시장 규모에서 35.5%를 차지하며 1위를 차지한 것으로 조사됐다. 특히 티맥스소프트는 작년 한 해 금융, 제조, 통신 등 다양한 산업분야에서 고른 성과를 이루면서 WAS 부분에서만 약 276억 원의 매출을 올려 지난 2007년과 비교해 약 15.8% 성장세를 보인 것으로 나타났다.

이러한 성과에 대해 한국IDC는 "티맥스소프트는 국내 최고 수준의 연구개발 인력은 물론 컨설턴트, 시스템 엔지니어 및 프로그래머 등 경쟁력 있는 기술인력을 보유하고 있어 국내 실정에 맞는 차별화된 컨설팅, 커스터마이징 서비스 제공 능력을 확보하고 있다"며 "과거 수년간 ASSP 시장에서 입증된 제품 성능과 안정성에 대한 고객들의 인식, 많은 레퍼런스 구축경험을 통해 확보한 경쟁력, 시장 리더십의 강점, 그리고 유연한 가격정책 등이 이번 성장에 기여한 것으로 분석 된다."고 밝혔다.

티맥스소프트 문진일 대표는 "핵심 시스템 소프트웨어 분야인 WAS 시장에서 국내 기업이 6년 연속 시장 1위를 달성했다는 것은 한국 소프트웨어 산업적으로도 역사적인 쾌거"라며 "이번 조사결과를 통해 티맥스소프트는 앞으로도 국내 WAS 시장에서 1위 독주 체제를 굳건히 하기 위해 노력해 나갈 것"이라고 말했다.

지난 2000년 티맥스소프트가 WAS 시장에 처음 선보인 '제우스(JEUS)'는 출시 3년 만인 2003년 오라클과 IBM 등 내로라하는 글로벌 기업들의 제품을 제치고 국내 시장점유율 1위에 올라섰으며, 이후 매년 격차를 벌려가며 지금까지 국내 WAS 시장의 최강자로 선두 자리를 지켜오고 있다.

끊임없는 연구개발 결과를 바탕으로 지난 2003년 '제우스 5.0' 제품이 자바 개발 국제 표준인 J2EE1.4 인증을 받은데 이어 2006년 5월에는 '제우스(JEUS) 6'로 Java EE 5(Java platform Enterprise Edition 5) 인증을 획득함으로서 한국 소프트웨어 기술력을 전 세계에 알리는 쾌거를 이루기도 했다.

'제우스'는 제품 출시 이후 지금까지 약 1,400여 곳의 고객사를 확보했으며 2008년 한 해 동안 농협중앙회, 우리은행, 조흥은행, NH투자증권, 행정안전부, SK텔레콤, 포스코, 삼성전자 등 600여 고객사에 공급됐다.

(4) 공개SW 기반 WAS 기술적 효과

□ 빠르고 유연한 개발

공개SW 기반 WAS 개발은 국내에서 상용으로 판매되는 티맥스소프트의 제우스가 제품의 인지도와 시장 점유율을 높이면서 국내 소프트산업의 가능성을 열었다. 하지만 공개SW 기반의 WAS 개발은 JBoss와 글래스피시, 그리고 아파치 톰캣을 커스터마이징하는 단계에 와 있으며, 공개SW 기반의 기술적인 개발은 아직은 요원하다.

□ 기술 자립 기반 확보

공개와 정보의 공유를 기반으로 하는 공개SW 기반 WAS의 개발은 시스템 소프트웨어에서 핵심 기술로 평가되고 있으며, 국내에서 아파치 톰캣이나 JBoss 기반의 다양한 애플리케이션이 개발되면서 점차 WAS에 대한 기술 자립도가 증가하는 추세에 있다.

□ 표준화 주도를 통한 선진 기술 시장 진입

웹 2.0과 웹 3.0 시대를 맞이하여 국내 연구 기관과 유관기업들이 국제 표준 단체에서 활동하면서 국내의 독자적인 기술을 기반으로 하는 표준을 선도하는 위치에 서게 되면서 선진 기술 시장에 진입하고 있다.

□ 핵심 인력 양성을 통한 기술 격차 해소

웹 2.0과 웹 3.0의 다양한 표준을 기반으로 다양한 교육이 이루어지며, 웹의 핵심 기술로 평가되는 AJAX, RSS, SOAP 등에 대한 기술적인 교육이 지속적으로 이루어지면서 웹 기반 기술의 격차를 줄이고 있다.

□ 기술 독과점 방지

기업 환경에서 안정적으로 평가되는 상용 WAS의 도입이 점차 공개SW 기반으로 진행되면서 기술적인 독점이 사라지고 있는 상황이지만 아직 공개SW의 신뢰성과 안정성에 대한 부분이 미약하므로 핵심적인 서비스에서는 기술적인 독점이 존재하고 있다.

마. DBMS

(1) DBMS 기술 동향

□ 특화된 DBMS 출현

- XML DBMS
- Main Memory DBMS
- Mobile DBMS
- 보안 강화 DBMS

□ 새로운 컴퓨팅 환경

- 그리드 컴퓨팅
- 자율 컴퓨팅
- 유비쿼터스 환경

(2) 상용 DBMS와 공개 DBMS 비교

(가) 상용 DBMS, 오라클

□ 오라클 DBMS 개요

미국 오라클사의 관계형 데이터베이스 관리 시스템(RDBMS)의 이름. 유닉스 환경에서 사용되는 RDBMS로는 현재 가장 널리 사용되는 대표적인 제품의 하나이다. 검색이나 업데이트용 언어로는 국제표준화기구(ISO)에서 표준화한 구조화 조회 언어(SQL)가 표준으로 되어 있다.

□ 오라클서버

- Oracle7 Enterprise Server : 가장 널리 사용되는 기업용 오라클 서버
- Oracle7 Workgroup Server : 워크그룹용 오라클 서버
- Personal Oracle7 : 개인용 오라클 서버
- Oracle RDB : DEC 플랫폼 전용 데이터베이스 서버
- Oracle Express : OLAP전용 데이터베이스 서버

□ 오라클 게이트웨이

비-오라클 데이터를 오라클 데이터와 같이 사용할 수 있는 전산환경 통합용 소프트웨어이다.

□ 인트라넷 솔루션

오라클은 기존의 오라클정보를 웹상에서 활용할 수 있도록 하여 웹 정보와 애플리케

이션에 꼭 필요한 정보를 제공합니다. 또한 타 DB 및 기존의 MIS정보를 통합 웹 환경에서 보다 적은 비용으로 사용 가능케 한다.

□ Oracle Web Server
Oracle Web Listener, Oracle Web Agent와 Oracle7 Server로 구성된 통합제품으로 개발자는 Web 애플리케이션을 Oracle7의 Stored Procedure와 같이 개발할 수 있도록 해줌으로 Web 애플리케이션 환경에 확장성과 이식성, 관리기능성을 부여해주고 온라인 환경에서 실시간으로 입력 및 수정되는 최신정보를 WEB사용자들에게 제공하여 실시간 정보공유를 가능케 한다.

□ Oracle Power Browser
강력한 웹기반의 클라이언트/서버 애플리케이션을 구축하기 위한 개발환경을 비롯, 최상의 Browser에서 필요한 모든 기능을 갖춘 웹 브라우저이다. HTML Document 작성 시스템과 개인용 Web Server뿐만 아니라 베이직 프로그램이 가능하며 확장성이 뛰어난 데스크톱용 개발환경으로 개인 또는 워크그룹 또는 기업 수준의 Web 애플리케이션 설계, 실행 및 유지관리를 할 수 있게 해준다.

(나) 공개 DBMS, MySQL

□ 개요
MySQL은 600만 사용자를 확보하고 있는 다중 스레드, 다중 사용자 형식의 구조질의어 형식의 데이터베이스 관리 시스템(SQL DBMS)이다. MySQL AB가 관리 및 지원하고 있으며, Qt처럼 이중 라이선스가 적용된다. 하나의 옵션은 GPL이며, GPL 이외의 라이선스로 적용시키려는 경우 전통적인 지적재산권 라이선스의 적용을 받는다. 위와 같은 지원 방식은 자유 소프트웨어 재단이 프로젝트에 저작권을 적용하는 방법과 비슷한 JBoss의 모델과 유사하다. 그러나 기반코드가 개인의 소유자에게 저작권이 있고 커뮤니티에 의해 개발되는 아파치 프로젝트와는 다르다.

MySQL AB는 MySQL 라이선스에 의한 판매 지원 및 서비스 계약 시스템을 개발, 유지한다. 그리고 또한 인터넷을 통한 전 세계의 협력자들을 고용한다. MySQL AB는 David Axmark, Allan Larsson 그리고 Michael "Monty" Widenius에 의해 설립되었다. MySQL AB는 또한 MaxDB라고 불리는 MySQL AB와는 기반코드가 다른 데이터베이스 관리 시스템을 판매하고 있다.
최근 썬 마이크로시스템즈에 85억 달러에 인수되었다.

(3) 공개SW 기반 DBMS 분야 국내 개발 동향

1) 국내 DBMS 기술 동향

한국IDC가 최근 발간한 "데이터베이스 관리 소프트웨어(DBMS) 시장 분석 및 업데이트 보고서, 2008-2012"1 에 의하면 2007년 국내 DBMS시장은 전통적인 DBMS 수요와 함께 비즈니스 관점에서의 실시간 정보 활용 지원 , 효율적 데이터 관리 등의 이슈에 힘입어 전년 대비 7.3%의 성장률을 보이며2,493억 원대 규모를 형성하였다.
최근 KRG 2의 조사에 따르면 오픈소스 DBMS의 증가, 유틸리티 성향의 라이선스 정책의 확대 등이 성장에 다소 부정적인 영향을 미치겠지만, 금융, 통신, 대형 제조업체에서 공공, 유통, SMB 등의 업종으로 확대되면서 성장세는 지속될 것으로 전망된다. 한편, 업종별로 살펴보면 제조와 유통 물류 업종의 신규도입 비중이 금융 및 서비스 업종에 비해 상대적으로 높게 나타났다. 제조업종의 경우 글로벌 시스템 통합 및 IT고도화, PLM, RFID, MES 등 생산정보화 구현, 유통/물류 업종의 SCM 및 RFID 구축 등이 DBMS 도입을 이끌고 있는 것으로 분석된다.3 눈에 띄는 것으로 공공시장을 중심으로 국내 업체들의 선전을 하고 있다는 것이다. 국내 DBMS 업체들은 신제품 출시와
공공 시장에서의 DBMS 분리발주, 유지보수 요율을 둘러싼 오라클과 고객과의 갈등, 그리고 모바일 DBMS 시장 활성화 등의 요인을 잘 활용하여 시장 확대에 적극적으로 나선다는 계획이다. 전체 시장의 90%(공공 시장의 93%)의 점유율을 차지하고 있는 오라클, IBM, Sybase 등 외국계 기업에 맞서 시장 점유율 10% 이상 달성을 목표로 하고 있다.
최근에 "NHN이 국산 DBMS 전문업체인 '큐브리드'를 인수하여 이를 오픈소스로 공개하겠다."는 내용은 오픈소스 DBMS의 증가추세에 있음을 시사한다. 실제로 IDC의 자료에 따르면 2008년 서버, 데스크탑, 패키징 시장에서 리눅스 진영의 총 매출은 357억 달러를 달성할 것으로 예상하였고, 이는 2006년 전 세계 DBMS 시장 총 매출의 2배가 넘는 것이다. 또한 가트너에 따르면 오픈소스 DBMS 시장은 2007년의 2억 달러에서 2008년에는 2억700만 달러 규모로 성장할 것으로 예상하였다.

2) 큐브리드(CUBRID)

□ 개요

큐브리드는 진정한 의미에서 관계형 데이터 모델과 객체지향 모델을 통합한 최초의 상용 DBMS라 할 수 있다. 많은 관계형 DBMS가 시스템 설계는 관계형으로 시작하여 엔진 구조를 수정하지 않은 채 객체지향 모델링 기능만 차후에 추가한 것과는 달리, 큐브리드는 처음 설계부터 완벽한 관계형 모델과 객체지향 모델을 동시에 지원할 수 있도록 시작하였다. 큐브리드는 SQL/X 언어를 지원하는데, 이는 SQL의 확장으로서,

기존 SQL의 관계형 기능을 모두 포함하면서 객체지향 기능을 추가한 형태이다. SQL/X는 데이터 정의 언어(data definition language), 질의(query) 및 데이터 조작 언어 (data manipulation language), 그리고 데이터 제어 언어 (data control language)를 포함하는 완전한 데이터베이스 언어이다. 큐브리드 시스템은 사용자의 질의나 데이터 조작 언어를 분석하여 최적의 수행 방법을 자동적으로 찾아낸다. 다중 사용자 환경에서 필요한 동시성 제어(concurrency control)와 복구 (recovery) 기능을 지원한다. 큐브리드의 주요 기능 중 하나는 멀티미디어 데이터와 장치(입력, 출력, 저장)를 관리하는 확장 프레임워크이다. 이 멀티미디어 프레임워크는 많은 관계형 DBMS가 지원하는 BLOB 기능이나, 혹은 몇몇 객체지향 DBMS가 지원하는 객체지향 프레임워크보다 훨씬 사용자 응용 요구에 가까이 다가선 것이다.

□ 특징

○ 대용량

큐브리드는 엔터프라이즈 데이터베이스로 멀티 볼륨을 지원을 통한 대용량 데이터베이스 환경을 제공한다. 데이터베이스 개수와 크기에 제한이 없으며, 각 데이터베이스에 생성 가능한 클래스(테이블)의 개수와 크기에 제한 없는 대용량 데이터베이스 환경을 제공한다. 각 클래스는 최대 6,400개의 열(컬럼)을 가질 수 있고, 각 열은 최대 2GB의 크기를 지원한다. 또한 최대 색인 개수는 6,400개로 대용량 데이터베이스 환경을 제공한다.

○ 확장성

멀티 볼륨 지원으로 볼륨 추가 및 관리기능을 제공하여 데이터베이스 저장 공간의 확장성을 제공한다. 또한, 멀티 쓰레드 서버 구조로 멀티 CPU 사용 서버의 성능을 최적화 하여 서버 확장성에 대응한다. 분산 데이터베이스 서버, 복제, 분할 지원으로 대용량 데이터베이스의 확장성을 최대화 하여 데이터베이스 환경 변화에 따른 유연한 대응 방법을 다양하게 확보하고 있다.

○ 고성능

▪ 고성능 서버 구조

큐브리드 데이터베이스 서버는 멀티 서버를 지원하고, 각 데이터베이스 서버는 멀티 쓰레드로 동작하도록 구성되어 동시 사용자 환경에서 고성능을 보장한다. 또한, 질의 수행 계획을 비용 기반 최적화 기법을 통해 작성하고 작성된 질의 수행 계획을 캐시하여 재사용성을 높임으로도 질의 수행의 성능을 극대화 하고 있다. 저장 구조적으로 분할과 복제를 제공하고, 질의 분할 최적화와 질의 분산을 통해 성능 극대화 기능을 제공한다.

▪ 고성능 클라이언트 구조
미들웨어인 큐브리드 브로커는 쓰레드 풀, 커넥션 풀을 관리 하는 것은 물론, 서버 부하를 자동으로 최적화는 기능을 제공하여 클라이언트의 성능을 향상시켜 준다. 또한 사용된 객체를 클라이언트 영역에서 캐시하여, 변경 점이 발생하지 않은 객체는 서버 질의 없이 클라이언트 사이드에서 재사용할 수 있어 성능을 극대화 할 수 있다.

○ 트랜잭션
큐브리드 데이터베이스 시스템은 트랜잭션의 원자성(Atomicity), 일관성(Consistency), 지속성(Durability), 독립성(Isolation)을 완벽하게 보장하기 위해 다음의 기능들을 충실하게 지원한다.
▪ 트랜잭션 단위의 commit, rollback, savepoint 지원.
▪ 시스템 / DB 장애 시 트랜잭션 일치성 보장.
▪ 복제간 트랜잭션 일치성 보장
▪ 다중 단위 잠금(multiple granularity locking) – DB, 테이블, 객체(레코드) 지원.
▪ 교착상태(deadlock) 자동 해결
▪ 분산 트랜잭션(2PC) 지원

○ 고가용성
백업/복구 기능에서 트랜잭션 일치성 보장하고, On-line 백업 지원 및 3단계의 백업 레벨 제공(전체, 1차 증분, 2차 증분)하여 백업의 효율성 및 관리의 편의성을 제공하고, 장애 발생 시점 또는 특정 시점으로의 복구를 지원한다. 큐브리드 7.0부터는 병렬 백업 및 실시간 압축 백업을 지원하여 백업의 효율성과 성능을 극대화하였으며, 기타 다양한 3rd-party 백업 솔루션과 직접 연동되어 있다. 또한 트랜잭션의 일치를 보장하는 마스터/슬레이브 방식의 복제 기능을 제공함으로써 고가용성을 실현하고 있다.

○ 보안 및 권한 관리
사용자, 그룹, 객체의 트랜잭션 별 권한을 사용자 별로 설정 가능한 권한 관리 기능을 제공한다.

○ 개발 생산성
▪ 표준 API
큐브리드는 C/C++를 위한 CCI API, DB C API부터 자바용 JDBC 및 TCL, PHP, Python, Ruby API를 제공한다. 또한 윈도우의 개방형 표준 인터페이스인 ODBC, OLE DB에 이르기까지 개발에 필요한 표준 API를 제공하며, C/C++ 내장 인터프리터인 ESQL/X를 제공함으로 어떠한 환경에서도 큐브리드를 사용하는 응용 프로그램을

작성할 수 있는 생산성을 제공한다.

▪ 자바 기반 저장 함수/프로시저

자바 기반 저장 함수/프로시저를 지원하여 JDBC 클라이언트 응용을 거의 수정 없이 그대로 서버 저장 프로시저로 전환할 수 있는 구조를 지원함으로써, 자바의 생산성과 재사용성을 높여주고, 자바 언어의 견고함과 유연함을 데이터베이스 응용에서도 그대로 사용할 수 있다.

▪ 관계형 모델링

ISO/ANSI SQL 92 표준을 충실하게 지원하며 풍부한 자료형을 제공하고, 각종 제약절 및 뷰와 트리거를 지원하여 원활한 관계형 데이터베이스 설계 및 개발, 운용을 지원한다.

▪ 객체지향형 모델링

클래스(테이블)간 상속(inheritance), 메서드, 사용자 정의 데이터 타입, 중첩 객체 (nested object), 집합형 데이터 타입(set, sequence, multiset)을 지원하여 객체지향적인 데이터베이스 설계 및 개발, 운용을 지원한다.

▪ 멀티미디어 프레임워크

멀티미디어용 데이터 타입과 메서드를 내장하여 부분 갱신 및 랜덤 접근이 가능하고, 데이터 저장 방법에 관계없이 동일하게 사용할 수 있는 프레임워크를 제공하고 있다.

▪ Large Object

데이터베이스 볼륨 내에 데이터 저장이 가능하며, 완전한 트랜잭션(동시성 제어, 백업 및 복구)을 지원하는 LO(Large Object)를 제공한다. 파일 시스템으로 데이터 저장하여 다른 프로그램과 데이터를 공유할 경우에도 동시성 제어를 지원하는 FBO(File based Object)를 제공한다.

○ 관리 편의성

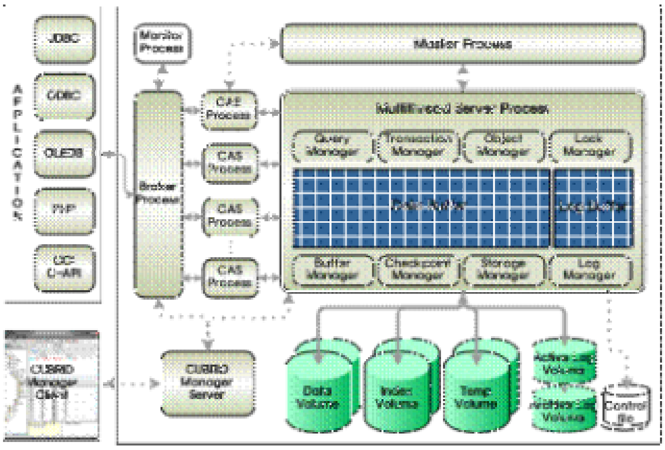
▪ 큐브리드 매니저

순수 자바로 작성된 통합 관리 툴인 큐브리드 매니저를 제공하여 플랫폼 독립적으로 관리, 질의, 진단, 튜닝을 하나의 툴에서 수행 할 수 있는 편리한 관리환경을 제공한다.

▪ 큐브리드 구조

큐브리드 데이터베이스 시스템은 객체 관계형 데이터베이스 시스템으로 데이터베이스

서버, 사용자의 질의 처리 요구를 담당하는 브로커, 데이터베이스 관리와 브로커 관리 그리고 진단 기능을 제공하는 큐브리드 매니저로 구성된다.



[그림 4-2. 큐브리드 구조]

(4) 공개SW 기반 DBMS 분야 기술 효과성

□ 빠르고 유연한 개발

공개SW 기반의 대표적인 MySQL DBMS의 공개 및 기술 공유를 통해 국내 업체들의 DBMS 관련 기술은 비약적으로 발전하고 있으며, 대표적으로 인터넷 최적화 기반의 큐브리드가 공개SW 기반으로 등장하면서 유연한 기술 개발이 가능해졌다.

□ 기술 자립 기반 확보

상용 DBMS인 사이베이스, 알티베이스, 티맥스소프트, 오라클과 SUN이 지원하는 대표적인 공개SW 기반의 MySQL이 점유하고 있는 시장에서 공개SW 정책으로 인터넷 최적화 DBMS인 큐브리드를 개발 및 기술지원

□ 표준화 주도를 통한 선진 기술 시장 진입

DBMS의 표준 언어인 SQL 기반으로 개발된 DBMS의 다양한 형태가 출현하면서 표준화에 대한 이슈가 등장하고 ANSI, SPAR 표준에 대한 작업이 시작됨

□ 핵심 인력 양성을 통한 기술 격차 해소

오라클 DBMS 기반으로 다양한 교육이 이루어지면서 DBMS에 대한 기술 습득이 가능해지고 동시에 국내 공개SW 기반 큐브리드의 기술 세미나 등을 통해 공개적인 교육 및 기술 전수가 진행되고 있다.

□ 기술 독과점 방지

오라클 주도의 DBMS 시장에서 공개SW인 MySQL과 국내 대표 공개SW인 큐브리드의 등장은 특정 벤더에의 종속성을 탈피하고 기술적인 독점 폐해도 줄이는 계기가 되었다.

바. IDE (Integrated Development Environment)

(1) IDE 기술 동향

임베디드 소프트웨어의 종합적인 개발환경은 타깃에서의 동작환경과 호스트에서의 개발환경으로 크게 나눠지며 실제로 적용할 수 있는 개발환경을 구축하려면 다양한 동작환경과 개발환경의 조합이 매우 다양하여 복잡하다. 타깃에서의 동작환경에는 타깃의 운영체계의 종류, 임베디드 프로세서를 포함한 하드웨어 플랫폼 등이 매우 다양하다. 임베디드 소프트웨어의 개발도구는 주로 PC/Windows 등 개발시스템에서 동작하지만 교차개발도구이므로 타깃의 동작환경과 밀접한 연관이 있다.

이러한 임베디드 소프트웨어의 개발환경의 다양성으로 처음에는 타깃의 운영체계와 개발도구가 고유의 프로토콜로 연동되어 분리할 수가 없었다. 예를 들어 VxWorks/Tornado는 실시간 운영체계인 VxWorks와 그와 같이 연동되는 개발도구인 Tornado가 분리될 수 없는 한 패키지로 제공되었다.

다음으로는 타깃의 운영체계와 개발도구가 분리하여 일정한 인터페이스만 구현하면 서로 연동이 가능한 구조가 되었다. 교차디버깅 전문 도구인 SingleStep, Code Warrior 등이 그러한 구조이다. 국내의 ETRI에서 주관하여 개발한 Esto도 그러한 분리형 구조를 갖는 대표적인 개발도구이다. 이러한 분리형 구조는 다양한 임베디드 소프트웨어의 환경에 쉽게 적용할 수 있는 장점이 있다.

최근에는 MicroSoft의 Visual Studio에 맞서 누구든지 기능을 쉽게 추가할 수 있는 개방형 구조(Open Architecture)의 공개소스형 소프트웨어 개발도구 협의회를 구성하였다. 2002년부터 IBM, Borland, Fujitu, Hitachi, Montavista, ETRI, HP 등이 참가하여 Linux상에서 Java를 기반으로 한 종합개발도구를 정의하고 필요에 따라 기능을 쉽게 추가할 수 있는 Plug-in 구조를 정의하고 개발하였다. Eclipse는 Java를 이용하여 개발되고 Java언어를 기본으로 지원하나 특별히 언어에 의존성이 적도록 설계하였다.

현재 Eclipse기반으로 플러그인 도구인 CDT(C/C++Development Tools)를 공개 프로젝트로 개발하여 많은 회사들이 이를 임베디드 SW개발도구로 확장을 하여 자사상품으로 내놓고 있다. TimeSys의 TimeStorm, Montavista의 DevRocket, WindRiver의 Workbench가 그것이다. MicroSoft의 Platform Builder는 Windows환경에서의 대표적인 개발환경인 Visual Studio의 축소판으로 임베디드 SW개발도구이다.

(2) 상용SW 기반 IDE와 공개SW 기반 IDE 비교

(가) 상용SW 기반 IDE

1) MS Visual Studio 2008

□ 개요

Microsoft® Visual Studio 2008은 개발자들이 최고의 품질과 풍성한 사용자 체험을 갖게 하는 연결 애플리케이션을 빠르게 제작할 수 있도록 하여, 스마트 클라이언트 애플리케이션에 대한 마이크로소프트의 비전을 이끌어준다. 조직들은 Visual Studio 2008을 통해서 비즈니스 관련 효율적 결정을 위해 정보를 얻고 분석하는 것이 기존보다 용이해졌으며 조직의 규모와 상관없이 Windows Vista 와 2007 Office 시스템의 장점을 이용해 보다 보안이 강화되고, 관리하기 쉬우며, 안정적인 애플리케이션을 빠르게 제작할 수 있도록 도와준다.

Visual Studio 2008 이 개발자들에게 제공하는 3가지 기본적 주요 발전사항은 다음과 같다.

- 빠른 애플리케이션 개발
- 효과적인 팀 협력
- 사용자 편의 극복

Visual Studio 2008은 다양한 플랫폼에 걸쳐 미래의 최첨단 애플리케이션을 빠르게 개발할 수 있는 혁신적인 기능, 데이터베이스 기능성, 디버깅 기능, 그리고 향상된 개발 도구들을 제공하며 .NET Framework 3.5를 이용한 보다 빠른 개발을 위한 비주얼 디자이너라든가 모든 형태의 데이터 개발을 빠르게 해주는 웹 개발 도구와 언어에 대한 중요한 개선사항을 포함하고 있다.
또한 Visual Studio 2008 은 개발자들이 능률적이고 표현이 풍부하며 AJAX 가능 웹 애플리케이션을 생성하기 위해 요구되는 모든 도구 및 framework 지원을 제공한다. 개발자들은 이러한 풍성해진 클라이언트측 및 서버측 framework 혜택을 이용하여, 어떤 후위(back-end) 데이터 provider 와도 통합될 수 있으며, 최신 브라우저 어디에서나 작동하고, 또한ASP.NET 애플리케이션 서비스 및 마이크로소프트 플랫폼에 완전히 액세스되는, 클라이언트 중심 웹 애플리케이션을 쉽게 개발할 수 있다.

○ 빠른 애플리케이션 개발

Visual Studio 2008은 개발자들이 현대적 소프트웨어를 빠르게 제작할 수 있도록, LINQ (언어 통합형 쿼리)와 같은 향상된 언어 및 데이터 기능을 가져오고, 이는 개별적인 프로그래머들이 정보를 분석하고 대응하는 솔루션을 개발하는 것을 보다 더 쉽

게 해준다.

또한, Visual Studio 2008은 개발자들이 동일한 개발 환경 내에서 서로 다른 복수의 .NET Framework 버전을 타깃으로 하는 기능도 제공하며 이에 따라 개발자들은 .NET Framework 2.0 이나 3.0 또는 3.5를 타깃으로 하는 애플리케이션을 만들 수 있으며, 이는 동일한 환경 내에서 다양하고 폭넓은 프로젝트를 지원이 가능하다.

○ 사용자 편의 극복
Visual Studio 2008은 웹, Windows Vista, Office 2007, SQL Server 2008, Windows Server 2008 등의 최신 플랫폼에서 연결 애플리케이션의 제작 속도를 높여주는 새로운 도구를 제공한다. ASP.NET AJAX 와 다른 신기술들은 개발자들이 웹에서 보다 효율적이고 쌍방향으로 개별화된 웹을 경험하는 새로운 세대를 빠르게 만들어나가도록 도와줄 것이다.

○ 효율적인 팀 협력
Visual Studio 2008은 데이터베이스 전문가와 그래픽 디자이너를 개발 프로세스로 통합하는 도구를 포함하여, 개발 팀의 협력증진을 확대 및 발전시킨다.

□ 특징
○ 전체적인 개발자 환경의 향상
Visual Studio 2008에서는 주요 제품의 품질을 향상시켰을 뿐만 아니라 많이 사용되는 디자인 화면에서 오류를 사용자에게 반환하는 방식을 변경하여 전체적으로 향상된 개발자 환경을 제공한다. 또한 다양한 .NET Framework 플랫폼을 대상으로 애플리케이션을 개발할 수 있다.

○ 새로운 웹 환경의 제공
효율적인 대화형 웹 애플리케이션을 보다 쉽게 작성할 수 있다. 친숙한 ASP.NET AJAX 프로그래밍 모델과 매끄럽게 통합하여 보다 효율적인 클라이언트측 실행이 가능하므로, 최종 사용자에게 빠르게 응답하는 웹 인터페이스를 제공할 수 있다.

○ Windows Vista 애플리케이션의 개발
새 WPF(Windows Presentation Foundation) 기능을 기존 Windows Forms 애플리케이션 및 새로운 애플리케이션 개발에 간편하게 통합함으로써, 향상된 플랫폼 기술을 쉽게 활용하고 보다 뛰어난 사용자 인터페이스를 가진 차세대 애플리케이션을 고객에게 제공할 수 있다.

○ 스마트 클라이언트 애플리케이션의 개발

Visual Studio 2008에서는 스마트 클라이언트를 쉽고 빠르게 개발할 수 있는 방법을 제공한다. 스마트 클라이언트를 기존 또는 새로운 웹 애플리케이션과 간단히 통합할 수 있는 종합적인 도구 및 클래스 집합을 제공하며, 연결이 불가능한 경우를 위한 로컬 데이터 캐싱을 지원한다.

○ Office 애플리케이션의 개발

VSTO(Visual Studio Tools for Office)는 Visual Studio 2008에 완전히 통합되었다. Outlook 및 PowerPoint와 같은 다양한 Office 애플리케이션을 개발하여 간편하게 배포할 수 있다.

○ 보다 생산적인 데이터 처리

LINQ(통합 언어 쿼리) 및 기타 향상된 데이터 액세스 방법이 도입되어 데이터를 처리할 수 있는 일관성 있는 프로그래밍 접근 방식을 적용하고, 새로운 데이터 디자인 화면에서 데이터 액세스를 수행한다.

○ ALM(애플리케이션 수명주기 관리)의 향상

Visual Studio 2008의 ALM 기능은 전체 소프트웨어 개발 수명주기 관리뿐 아니라 엔터프라이즈 애플리케이션의 최종 사용자 및 IT 관리자와의 중요한 상호 작용에 대한 탁월한 지원을 제공한다.

(나) 공개SW 기반 IDE

1) 이클립스(Eclipse)

□ 개요

이클립스(Eclipse)는 다양한 플랫폼에서 쓸 수 있으며, 자바를 비롯한 다양한 언어를 지원하는 프로그래밍 통합 개발 환경을 목적으로 시작하였으나, 현재는 OSGi를 도입하여, 범용 응용 소프트웨어 플랫폼으로 진화하였다.

자바로 작성되어 있으며, 자유 소프트웨어이지만 막강한 기능을 자랑한다. 2004년과 2005년 졸트 어워드(Jolt Award)를 수상했다. 식(蝕)을 뜻하는 이클립스란 이름은 자바를 개발한 회사의 Sun이란 이름을 염두에 둔 것으로 보인다.

원래 IBM의 WebSpheare Studio Application Developer란 이름으로 개발되었던 것인데, 엔진부분을 오픈소스로 공개한 것을 기반으로 지금의 이클립스로 발전해왔다.

최근 버전은 3.5로, 각 목적별(C/C++ 개발자용, 자바 개발자용, 웹 개발자용 등등)의 다양한 배포판이 존재한다.

□ 아키텍처

이클립스의 기반은 리치 클라이언트 플랫폼 (RCP)으로, RCP는 아래와 같은 구성 요소로 이루어져 있다.

- 코어 플랫폼 - 플러그인을 실행하는 이클립스의 기반
- OSGi - 표준 번들링 프레임워크
- Standard Widget Toolkit (SWT) - 이식 가능한 GUI 위젯 툴킷
- JFace - SWT를 이용하여 고급 컴포넌트를 구성하는 작업을 간략화 할 수 있도록 해준다.
- 이클립스 워크벤치(Workbench) - 뷰(views), 편집기(editors), 퍼스펙티브 (perspectives), 마법사(wizards)

이클립스의 GUI 위젯은 자바를 위한 위젯 툴킷인 SWT로 구현되었다. 이는 AWT나 Swing을 사용하는 다른 대다수 자바 응용 프로그램과 다른 점이다. 또한 이클립스의 사용자 인터페이스는 또 하나의 중간 GUI 계층인 JFace를 사용하여, 이는 SWT 기반의 응용 작성을 좀 더 쉽게 만들어 준다.

이클립스는 RCP를 포함한 상위의 모든 기능을 제공하기 위해, 플러그인을 사용한다. 이는 다른 자바 응용이 기능 확장 부분을 하드 코딩하는 것과는 대조적인 특징이다. 이 플러그인 메커니즘은 경량화한 소프트웨어 컴포넌트 프레임 워크라 할 수 있다. 플러그인 프레임워크를 활용하면, 자바가 아닌 다른 프로그래밍 언어, 예를 들면, C 언어나 파이썬 언어로 이클립스를 확장하는 것이 가능할 뿐 아니라, 이클립스에서 LaTeX와 같은 조판 언어를 활용하거나, 텔넷과 같은 네트워크 응용을 실행하거나, 데이터베이스 관리를 하는 것도 가능하다. 또한 플러그인 아키텍처로 개발 환경 자체를 확장하는 것도 가능하다. 예를 들어 기본으로 제공하는 CVS 지원에 더욱 향상된 형상관리 기능을 확장하는 것이 가능하다.

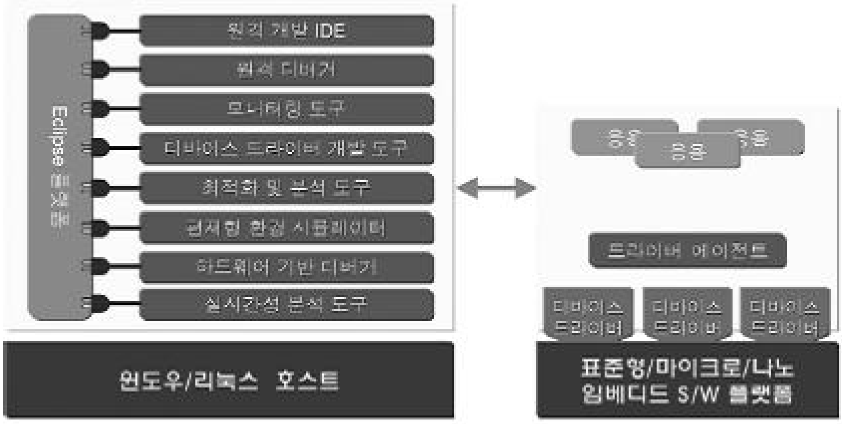
이클립스 SDK에는 이클립스 자바 개발 툴(Java Development Tools)이 포함되어 있는데, 이는 자바 컴파일러를 내장하고 있고, 자바소스 파일 전체 모델을 제공한다. JDT를 이용하면 자바 언어기반의 고급 리펙터링 기법과 코드분석을 수행할 수 있다.

- 개발자 : 이클립스 재단
- 최근 버전 : 3.5.0 / 2009년 6월 11일 수요일 17시 0분 (GMT -05:00)
- 운영체제 : 크로스 플랫폼
- 환경 : x86-32 비트, 자바
- 종류 : 통합 개발 환경
- 라이선스 : 이클립스 공중 허가
- 웹사이트 : http://eclipse.org

(3) 공개SW 기반 IDE 분야 국내 개발 동향

(가) Eclipse 기반의 ETRI 통합개발환경 (Eclipse ESTO)

ETRI에서는 2000년도 Eclipse 초기 년도부터 가입하여 2004년도부터 공개된 Eclipse 플랫폼과 CDT를 이용하여 임베디드 SW 개발환경을 본격적인 개발에 착수하였다. 그림 2는 Elipse기반 ETRI 통합개발환경(Eclipse Esto)의 내부 구성도이다. Eclipse Esto는 Eclipse 플랫폼에 임베디드 시스템 특성에 맞게 정의한 원 격 개 발 IDE(Integrated Development Environment), 원격 debugger, 타깃 event monitor 등을 개발하여 임베디드 SW 개발환경의 기초를 구축하고 디바이스 드라이버 개발도구, 최적화 및 분석도구, 하드웨어 기반 디버거, 실시간성 분석도구 등을 추가하여 임베디드 SW 개발의 통합도구로서 면모를 갖추고 있다.



[그림 4-3. Eclipse Esto의 구성도]

특히 최적화 및 분석도구는 소스 수준의 성능, 코드 크기 그리고 전력소모량까지 분석하도록 하여 임베디드 SW의 성능, 메모리 사용량 등의 최적화뿐만 아니라 유비쿼터스 환경에서의 전력소모량을 최소화하는 도구로 유용하게 사용될 수도 있다. Device driver 개발도구는 Linux를 기반으로 다양한 device의 종류에 따라 driver의 기본 구조 소스를 자동으로 생성하고 시험을 보조함으로써 각종 device driver개발을 용이하게 하고 있다. 또한 표준형, 마이크로형 및 나노형 Qplus 모두에 적용하도록 함으로서 동일한 환경으로 모든 개발환경에 사용되도록 설계되었다.

(나) Visual Esto의 상용화

Visual Esto는 ETRI와 (주)코스모의 공동 개발하여 (주)코스모에서 상용화하여 국내 교육기관 및 기업에 보급되고 있는 임베디드 SW 통합개발도구이다. Visual Esto는 국내에서는 취약한 기반 소프트웨어 분야인 임베디드 시스템 소프트웨어가 세계적인 경쟁력을 갖게 된 유일한 제품으로서 다음과 같은 특징을 갖는다.

□ Embedded 소프트웨어의 개발을 PC상에서 MS Visual Studio와 유사한 방법으로 용이하게 개발할 수 있다. 즉 초기에 임베디드 시스템의 특성만 정의해두면 개발자가 임베디드 시스템에 고유한 것을 몰라도 PC상에서 손쉽게 임베디드 SW를 개발할 수 있도록 하였다.

□ 임베디드 소프트웨어의 모든 단계, 즉 Linux 등 OS상의 응용 SW개발뿐만 아니라 OS내의 device driver 개발, on-chip debugging기술에 의한 firmware 개발 등 모든 단계에 세계 최초로 동일한 도구를 적용하도록 하였다.

□ 비정지 정지점 설정, 디버깅 도중 실시간 타겟 제어 가능, 실시간 attach기능 등 다양한 실시간 디버깅 기능이 실현되어 있어 임베디드 시스템의 중요한 특징인 실시간성으로 실시간 동작 중의 오류도 디버깅할 수 있다.

□ 실행환경(타겟/OS)와 개발환경(호스트/IDE)이 분리된 개방형 구조로서 동일한 도구로 Embedded Linux상에서의 대형 응용뿐만 아니라 소형 real-time OS 응용 등 어떤 종류의 임베디드 시스템도 수용할 수 있어 ubiquitous 환경에 적합하다.

□ 타겟 모니터 및 트레이서, 임베디드 profiler, 고급 memory debugger 등 풍부한 시스템 종합시험 및 분석도구를 갖추고 있어 복잡한 응용시스템의 시험 기간을 효과적으로 단축한다.

(4) 공개SW 기반 IDE 기술적 효과

□ 빠르고 유연한 개발

공개SW 기반인 Eclipse IDE를 기반으로 국내에서 개발된 Esto와 같은 통합 개발 환경 도구의 빠른 개발이 가능해지게 되었지만, 기반이 되는 플랫폼에 대한 기술은 아직 요원한 편이다. 하지만 점차 국내 연구 기관과 각 기업들이 공개SW 기반의 커뮤니티를 통해 IDE에 대한 기술 개발에 박차를 가하고 있다.

□ 기술 자립 기반 확보

통합 개발 도구의 개발은 많은 시간과 노력이 투입되어야 하지만 동시에 기반 기술이 없으면 개발하기 힘든 분야중의 하나이다. 하지만 국내 ETRI에서 개발된 Esto를 비롯하여 공개SW인 Eclipse를 통해 기반기술을 습득하고 있으며 이를 기반으로 기술 자립도가 향상되고 있다.

□ 표준화 주도를 통한 선진 기술 시장 진입

통합 개발 환경 도구는 선진 기술의 대표 주자이지만 국내에서 개발됨으로 인해 선진 기술 시장에 진입할 계기는 만들어졌지만, 국제적인 표준 언어와 표준화를 위한 다양한 커뮤니티의 활성화를 통해 선진 기술 시장에 진입이 가능하다.

□ 핵심 인력 양성을 통한 기술 격차 해소

통합 개발 환경 도구에 대한 기술 교육 및 핵심 인력 양성은 국가 정책 중의 일환인 임베디드 소프트웨어 인력 양성 사업의 일환으로, 그리고 국내 기업 환경에서 통합 개발 환경 도구에 대한 교육은 개발 주기를 빠르게 하는 부분이므로 주기적인 교육이 이루어진다. 사용자 교육 및 개발자 도구 활용 교육에 더하여 핵심 기술 기반의 교육으로 진행되어야 할 필요성이 있다.

□ 기술 독과점 방지

공개SW 기반 통합 개발 환경 도구의 등장과 구글을 포함한 세계적인 임베디드, 모바일 업체들이 자사의 통합 개발 환경 도구를 공개SW 기반의 도구를 기반으로 개발함으로써 기술 독과점이나 벤더 종속적인 부분은 많이 해소되었다.

사. Embedded System

(1) Embedded system 기술 동향

현재 빠르게 성장하고 있는 임베디드 SW(embedded software) 산업은 기술 혁신과 시장에서의 치열한 경쟁에 의해 새롭고 역동적인 양상으로 전개되고 있다. 임베디드 시스템(embedded system)은 일반적인 목적보다는 특수한 애플리케이션용으로 설계되기 때문에, 이 산업계는 자체 개발을 추구하는 수백 개의 소형 업체들이 난립하고 있다. SoC(System-on-Chip)의 출현으로, 산업계는 보다 복잡한 시스템 설계를 위해 역동적인 구조 변화를 일으키는 창조적인 새로운 솔루션을 필요로 하게 되었다. 많은 임베디드 SW 업체들은 임베디드 SW 요구에 대응할 원스탑샵(one-stop-shop)이 되기 위해 다른 업체를 적극적으로 인수/합병하고 있으며, 임베디드 HW 업체와 장비 업체들까지 임베디드 SW 산업에 진입하기 시작하여 더욱더 치열한 경쟁 환경이 조성되고 있다.

당초 산업용 기기를 제어하기 위해 사용되던 임베디드 SW는 공장 자동화 및 가정 자동화에 필요한 자동제어 시스템을 비롯해 각종 디지털 정보가전 기기, 자동 센서장비 등의 기능을 다양화하고 부가가치를 높일 수 있는 핵심 SW로 자리매김 하고 있다. 최근에는 임베디드 SW의 멀티미디어 처리, 다중작업 및 실시간 처리 능력이 강화되고 유무선통신 및 네트워크와의 접목으로 전통적인 제조, 유통, 금융, 서비스 산업뿐만 아니라 항공, 우주, 국방, 의료, 멀티미디어 통신 및 에너지 개발 등의 첨단 분야에 이르

기까지 그 사용범위와 영향력이 점점 커지고 있다.

(2) 상용SW 기반 Embedded 시스템과 공개SW 기반 Embedded 시스템 비교

(가) 상용 SW 기반 임베디드 시스템 소프트웨어

1) VxWorks

□ 개요

미국의 윈드리버 시스템사가 만들어 판매하는 실시간 운영체제(RTOS)인 VxWorks는 세계적으로 가장 많은 장치에 설치되고 탑재된 장치 소프트웨어이다. 최신 버전인 VxWorks 6.0에는 더 강력한 새로운 기능들을 첨가됐으며, 개방성, 성능, 신뢰성 그리고 상호운용성에 초점을 뒀다.

VxWorks 6.0은

▪ Posix와 IPv6와 같은 오픈 스탠다드를 통한 개발자의 생산성 최대화
▪ MMU 기반 메모리 보호를 통한 신뢰성 향상
▪ 향상된 에러 관리를 통한 시장 출시 촉진
▪ 오픈 소스를 포함해 현재 사용하는 VxWorks 혹은 다른 IP로의 간단한 마이그레이션
▪ 뛰어난 성능, 신뢰성, 판단력, 확장성, 낮은 대기시간을 포함한 VxWorks의 핵심 특성을 반영하는 제품출시의 지속성

상호보존성을 지닌 VxWorks 솔루션 요소들은 통합 개발도구 (Wind River Workbench), 보드 미들웨어의 적용성, 보드 프로세서와 BSP의 적용성, 윈드리버의 프로페셔널 서비스 지원을 포함할 뿐만 아니라 많은 VxWorks 개발 경험자들과 툴, 하드웨어 미들웨어와 어플리케이션 제공자들의 확장된 커뮤니티도 제공한다.

□ VxWorks의 특징 및 이점

○ 메모리 보호

- MMU 기반 메모리 보호
- 프로세서 기반 사용자 모드의 적용실행 환경
- 확장된 유저모드 시스템 콜 인터페이스
- 프로세스와 태스크를 위한 메모리 공유
- 프로세스를 위한 문서 공유
- 프로세서 간 교신을 위한 독립된 메시지 채널 선정

○ 에러 관리
- 태스크 초과 감지
- 코드 충돌 감지
- 불필요한 지표들의 사용 감지
- 힙 블록 초과 감지
- 힙 사용 추적과 누출 감지
- RTP 에러 감지
- 커널 적용 에러를 위한 API 지원
- 사용자 모드 적용 에러를 위한 API 지원
- 적용 가능한 에러 관리 수단
- 흔히 볼 수 있는 헤더 문제, 핵심 OS와 하드웨어 정보, 사용자 규정의 옵션 스트링을 포함하는 광범위한 에러 기록

○ Wind® 마이크로커널
- 효율적 태스크 관리
- 멀티태스킹, 태스크 수 무제한
- 256개의 우선순위
- 선점형 라운드 로빈 스케쥴링
- 신속하고 유연한 태스크간 교신
- 바이너리, 카운팅, 유텍스 세마포어 및 우선순위 계승
- VxWorks 이벤트
- 지역 및 분산 메시지 큐
- POSIX 파이프, 카운팅 세마포어, 메시지 큐, 시그널 및 스케줄링
- 컨트롤 소켓
- 동적 메모리 관리
- 향상된 베스트-핏 메모리 할당 알고리즘
- 자동 자원 재생
- 컴포넌트의 인크러멘틀 링킹 및 로딩
- 신속하고 효율적인 인터럽트 및 예외 처리
- 최적화된 부동소수점 지원
- 시스템 클락 및 타이밍 시설
- 확장성이 높은 메모리 사용량
- 고도로 유연한 디자인으로 광범위한 응용 가능

○ 윈드리버 네트워크 스택 2.0

- IPv6 대비 - 승인된 IPv4/IPv6 이중 스택 구현
- 하나 또는 복수의 사용자 모드 프로세스에서 실행하는 복수의 IPv4 및 IPv6 응용프로그램
- 원격 프로시저 호출 프로토콜 버전 2
- RSH, FTP, Telnet 및 TFTP 등의 원격 파일 접근 프로토콜
- DHCPv4 및 DHCPv6
- TCP 및 UDP에 대한 표준 BSD 소켓 인터페이스
- 자동 IPv4 구성
- fastpathlib 및 fastu의 성능 개선
- 유닉스 스타일의 네트워크 스택 구성 및 모니터링 유틸리티 구현
- RIP 및 RIPng 구현
- IPv4, UDP, TCP, ICMP를 위한 표준 MIB 및 SNMP 관리를 지원하기 위한 네트워크 인터페이스

□ 적용 사례

화성 탐사선 스피리트 호와 오퍼튜니티 호는 PowerPC 플랫폼에 VxWorks 운영체제를 탑재하였다. 이 밖에도 몇 기의 우주선의 운영체제로 사용되었고, 보잉사도 새로운 상업 항공기 기종인 7E7의 운영체제로 사용하려고 한다.

2) Windows Embedded CE

□ 개요

윈도 CE(Windows CE)는 소형 컴퓨터나 PDA 등에 사용되는 마이크로소프트 윈도 제품으로, 데스크톱의 커널을 변경하여 만들지 않고 처음부터 새로 만든 제품이다. 인텔 X86 및 호환 제품, MIPS, ARM, 히타치 SuperH 프로세서를 지원한다.

□ 특징

윈도 CE는 저장 공간이 충분하지 않은, 이를테면 개인용 정보 단말기나 모바일 장치 등에 최적화되어 있다. 윈도 CE 커널 자체만으로는 1 메가바이트 이하의 메모리에서도 동작이 가능하도록 설계되어 있다. 장치들은 디스크 저장 장치를 사용하지 않고 설정할 수 있으며, 사용자가 추가하지 못하도록 만들 수도 있다. 윈도 CE는 실시간 운영체제를 표방하고 있으며 256단계의 우선순위 정도를 가지고 있다. 유닉스 계열 운영체제와는 달리 실행 파일의 기초 단위는 스레드이다.

마이크로소프트는 "CE"에 어떠한 뜻도 없다고 밝혔다. 하지만 대부분 사람들은

"Consumer Electronics" 혹은 "Compact Edition"의 약자라고 생각한다. 마이크로소프트는 윈도 CE 디자인 목표 중 "Compact, Connectable, Compatible, Companion"을 암시한다고 하지만 절대로 특정 단어의 약자는 아니라고 밝혔다. 코드명 페가수스로 알려진 첫 번째 버전은 윈도와 비슷한 인터페이스를 채용했고, 마이크로소프트의 인기 있는 프로그램(MS Word, Excel 등)들이 이식되었다.

윈도 CE는 초기에 PDA의 운영체제로 주로 사용되었지만 AutoPC, 스마트폰 등의 기기에도 사용되었다. 어떤 윈도 CE 버전은 SEGA 드림캐스트에서 작동하고, 엑스박스 게임기에서 동작하는 버전도 있다.

많은 마이크로소프트 제품들과 달리 윈도 CE는 원시 코드 형태로 제공된다. 따라서 Windows CE는 제조사의 설정에 맞게 디자인하거나 커스터마이징이 가능하다. 우선 여러 하드웨어 업체들에게 소스 코드가 제공되어서 그들의 장치에 이식할 수 있도록 한다. 이를 기반으로, 윈도 CE 운영체제 이미지 작성 및 사용자 정의 도구인 Platform Builder(플랫폼 빌더) 같은 제품이 소스 코드를 포함한 형태로 공개된다. 하지만 이 과정은 상당히 복잡하고 문서화되지 않았다는 점에서 비판을 받고 있다.
Windows CE는 4.2 버전까지는 출시 이름 뒤에 .NET 이 붙었으나, 5.0 이후로는 Windows Embedded CE 로 명칭이 변경되었다.
현재 Windows CE는 6.0 R2 까지 출시된 상황이다.

□ 세부 정보
- 회사 / 개발자 : 마이크로소프트
- OS 계열 : 윈도 CE
- 소스 형태 : 클로즈드 소스
- 최신 버전 : 6.0 R2
- 최신 버전 출시일 : 2007년 11월 15일
- 사용 가능한 언어 : 다국어
- 지원되는 플랫폼 : x86, MIPS, ARM, SuperH
- 커널형태 : 모노리식 커널, 하이브리드식 커널
- 라이선스 : 마이크로소프트 EULA
- 웹사이트 : 윈도 CE

(나) 공개SW기반 임베디드 시스템

1) 임베디드 리눅스

□ 개요

- HW의 발전
- 32/64bit 고성능 CPU
- SW의 많은 기능 요구
- O/S의 기능이 중요, 안정된 운영체제의 제공(멀티태스킹)
- 강력하고 다양한 네트워크 환경의 제공
- 다양한 형태의 파일시스템과 실행파일 포맷 지원
- 확장성의 다양함과 용이함의 제공
- Free Software

□ 특징

- 기능성과 확장성이 우수 (리눅스 이용에 따른 장점)
- PowerPC, ARM, MIPS 등 다양한 CPU Platform지원함
- 로열티가 없으므로 가격 경쟁력이 우수
- 사용자층이 넓어 오류 수정이 빠르고 안정성이 우수
- 기존의 데스크 탑 개발 환경과 동일하여 개발이 용이함

2) TRON

□ 개요

도쿄대 교수인 Ken Sakamura 교수에 의해 네트워크화 된 유비쿼터스 컴퓨팅이라는 아이디어를 처음 제안한 것은 1984년도로서, 당시 그는 TRON(The Real-time Operating System Nucleus)이라는 개방 아키텍처를 고안해 냈다. 그 때 이후로 Sakamura 씨는 TRON 프로젝트와 여러 파생 프로젝트들을 이끌어 왔는데, 이는 모든 물건들이 컴퓨터를 포함하고 있으며 네트워크에 연결되어 있는 환경을 구현하기 위한 핵심 기술들을 개발하기 위한 것이었다. 예를 들어, T-Engine Forum은 2002년에 TRON을 중심으로 임베디드 개발을 촉진시키고자 설립되었다. 그 일년 후에는 Ubiquitous ID Center가 설립되었는데, 그 목표는 Sakamura 씨의 "ucode"를 중심으로 RFID 기술을 널리 보급시키는 것이었다. 유비쿼터스 컴퓨팅을 위한 인프라로서 설계된 code는 슈퍼마켓의 완두콩 캔에서 엠파이어 스테이트 빌딩에 이르기까지 어떤 것이든 그 물건 특유의 코드를 부착할 수 있다.

(3) 공개 임베디드 시스템 국내 동향

(가) 국내 임베디드 동향

임베디드 SW는 산업간 기술 융합에 의한 다양한 서비스를 시공간의 제약 없이 제공

하기 위해 전자제어기기의 고기능화, 고성능화 목적으로 내장된 소프트웨어이다. 즉, 제품을 고도화시키는 존재라 할 수 있다. 때문에 최종 제품의 부가가치를 결정하는 요소기술로서 포스트 PC 시대의 성장 가능성이 매우 높은 핵심 산업으로 떠오르고 있다. 또한 산업별 SW 개발비의 원가 비중도 지속적으로 증가하고 있는데, 평균 50% 이상의 비용이 SW 개발비로 소요되고 있다. 고급 항공기의 경우는 SW 개발 비용이 전체 개발비의 60%가 차지한다. 이런 개발 단계를 거쳐 제품은 계속적으로 첨단화되고 있는 것이다.

임베디드 시장 동향을 살펴보면 세계 임베디드 SW 생산액은 2012년까지 연평균 2.7%의 성장이 예상되고 있는 가운데, 국내 시장은 2012년까지 연평균 5.1% 성장에 이를 것으로 추정된다. 전 세계 시장 규모가 114조원 정도인 가운데 국내 시장이 8조4000억원 정도를 형성하고 있다, 이는 시장 규모로 봤을 때 전 세계 시장의 6%를 차지하는 것으로, 임베디드 SW 분야가 패키지 SW나 IT 서비스 산업 보다 많은 비중을 차지한다. 이는 국내 산업 중에서 제조업이 굉장히 활발히 이뤄지고 있음에 따라 시스템의 생산량도 높아진 것으로 보인다.

세계 시장에서 생산되는 임베디드 시스템은 2006년 44억 개에서 2009년 63억 개로 연평균 12.5%의 성장률을 보이고 있다. 또한 임베디드 SW 솔루션 공급업체로 거래되는 임베디드 디바이스는 과거 가전 중심에서 다양한 산업으로 확산되는 추세다.

개발 도구 시장의 경우 세계시장에서 생산되는 개발 도구 시장은 2005년 3억 달러에서 2009년 3억4000달러로 연평균 3%의 성장률을 보이고 있다. 지역별로는 아시아태평양지경의 개발 도구 시장이 가장 큰 성장을 하고 있다.

임베디드 OS 시장의 경우 인 하우스 개발이 많이 감소하면서 리눅스가 급격하게 증가할 것으로 업계전문가들은 전망하고 있다. 자동차, 가정, 의료기기 등 다양한 산업분야에 적용돼 시장은 계속적으로 성장할 것으로 예상된다.

임베디드 개발 인력은 2009년까지 2.3%로 증가할 것으로 예상되며, 특히 SW 개발 인력의 증가가 두드러지는 것과는 달리 하드웨어 인력은 감소하는 것으로 나타난다. 산업별로 보면 의료기기, 휴대폰, 자동차 분야 등에서 임베디드 SW 인력 수요가 증가할 것으로 예상하고 있다.

국내 임베디드 SW 시장 동향을 보다 구체적으로 살펴보면 국내 임베디드 SW 시장은 인 하우스 형태의 자체 개발이 87.5%, 용역 개발이 10.5%, 라이선스/로열티 개발이

2%로 구성된다.

산업별로 보면 세계 시장과 비교해 보면 먼저 자동차 전자제어 산업 시장에서는 하이브리드 자동차, 지능형 자동차 등의 공급으로 2012년까지 세계시장이 연평균 5.5% 성장하는 가운데, 국내 시장은 2배가량 높은 9.2%의 높은 성장이 예상된다. 향후 5년 후에는 자동차 내의 전장부품이 40%로 높아질 것으로 내다보고 있다.

정보가전 분야에서는 전 세계 시장이 가전제품의 포화와 세계 경제 둔화로 2012년까지 연평균 0.03%로 성장세가 둔화될 것으로 예상되며, 국내는 마이너스 성장까지 조심스레 예상되고 있다.

국방과 항공 산업에서는 2012년까지 세계시장이 연평균 0.17% 성장하고 국내 시장은 수출 등이 가시화돼 5.6%의 높은 성장을 할 것으로 전망된다. 특히 이 산업 분야는 실시간 처리, 고도의 신뢰성, 예측 가능성 및 조립성, 적응성 등의 섬세하고 정밀한 특징이 요구된다. 현재 빅5 업체의 매출액이 세계 100대 방산 업체 전체 매출액의 40% 이상을 차지하고 있다.

마지막으로 정보통신 산업 분야에서는 2012년까지 세계 시장은 연평균 5.95% 성장하고 국내는 12.8의 급성장이 예상된다. 휴대폰 및 차세대 이동통신 단말 시장을 중심으로 지속적인 성장이 예상되는 가운데, 3GPP, OMA, LIMO, OSGi 등 다양한 표준화 단체를 통한 기술 개발이 이뤄지고 있다.

(나) 임베디드 SW 표준 플랫폼의 기반이 되는 국산 임베디드 운영체제 Qplus

□ 개요

Qplus는 기존의 임베디드 운영체제와는 달리 유비쿼터스 컴퓨팅 환경에서 표준형, 초소형, 경성 실시간 임베디드 시스템에 활용될 수 있도록 네트워크 기반 동작과 유비쿼터스 네트워크 시스템을 지원한다. 또한 임베디드 운영체제에 다양한 미들웨어와 사용편의성이 강조된 개발도구를 추가한 임베디드 소프트웨어 표준 플랫폼으로 제공되고 있다. 임베디드 시스템은 수행 목적에 따라 다양한 규모와 특성의 임베디드 소프트웨어가 요구되는 반면, 응용 소프트웨어 프로그램 개발자들은 보다 일관성 있고 통일된 개발 체계를 원하고 있다.

Qplus는 이러한 요구사항을 모두 수용하기 위하여 임베디드 소프트웨어 표준 플랫폼 API를 먼저 정의한 후, 이를 준수하는 표준형·마이크로·나노의 3가지 다른 규모와 특성의 임베디드 플랫폼을 제공하고 있으며 이러한 특성으로 인해 차세대 소프트웨어

표준 플랫폼으로서 각광받고 있다. 임베디드 시스템은 크기·가격·자원 등의 하드웨어 구성이 제한되기 때문에 경량화, 저전력 지원, 자원의 관리 등의 하드웨어 최적화 기술이 포함된다. 뿐만 아니라 경량 네트워크 지원, 실시간 스케줄러, 멀티미디어 파일 시스템, 빠른 부팅 기능 등이 제공되며, 이러한 기능들을 타깃 임베디드 시스템에 쉽게 적재하기 위한 설정도구 등이 포함된다.
커널의 HAL(Hardware Abstraction Layer)은 다양한 하드웨어를 운영체제가 관리하기 위해 추상화하여 제공하는 계층이며, 타깃 빌더의 QSP(Qplus Support Package)는 운영체제 커널뿐만이 아니라 멀티미디어 플레이어, 자바 가상 머신 등의 미들웨어, 응용 프로그램 구성 및 설정을 위한 기능들도 제공한다.

□ 특징
- 실시간 지원
- 저전력 지원
- 빠른 부팅 지원
- 다양한 디바이스 드라이버 지원
- 플래시 파일 시스템(FFS) 지원
- 멀티미디어 파일 시스템(MMFS) 지원
- 최적화된 Footprint 지원
- 임베디드 GUI 지원
- Java 클래스 라이브러리 지원

□ Qplus 배포판 및 지원
Qplus 배포판으로는 Generic 버전인 Qplus-CE와 Qplus-ME가 있고, 특정 임베디드 시스템에 적용된 버전으로는 Qplus-CE를 기반으로 홈 서버에 적용한 Qplus-HS, 디지털 홈 엔터테인먼트 센터에 적용한 Qplus-EC가 있으며, Qplus-ME를 기반으로 한 시스템으로는 제주 텔레매틱스 시범 사업 단말에 적용될 예정인 Qplus-TM, 국민 로봇 시범사업의 URC 로봇 플랫폼에 적용될 Qplus-RB가 있다.
시스템 개발자 버전은 연구단 내 또는 외부 기술이전 업체의 시스템 개발자를 위해 만들어진 버전으로 GUI, 주로 디바이스 드라이버 개발 등 시스템 프로그램 개발 목적으로 릴리즈 되며 CDT, 개발 패키지(RPMS, SRPM), 타깃 루트 파일 시스템으로 구성되어 있다.
응용 프로그램 개발자 버전은 시스템 개발자 버전을 기반의 업그레이드 버전으로 Firefox, mplayer 등 응용 프로그램 개발과 JVM 등 미들웨어 등의 응용 프로그램을 개발하기 위해 만들어졌다. TB, ESTO, CDT, 최종 개발패키지(RPMS, SRPM), 타깃 패키지(최종 응용 프로그램 미포함), Eclipse 환경(JDK, JDT), 타깃 루트 파일

시스템으로 구성되어 있다.
최종 사용자 버전은 응용 프로그램을 추가하여 생성한 사용자를 위한 버전으로 유지·보수되는 에디션이다. TB, ESTO, CDT, 개발 패키지, 타깃 패키지(응용 프로그램 포함), Eclipse환경(JDK, JDT)으로 구성되어 있다.

(4) 공개SW 기반 임베디드 시스템 기술적 효과

□ 빠르고 유연한 개발
SW 개발 사이클의 유연성과 개발 과정의 학습 효과

□ 기술 자립 기반 확보
대표적인 공개SW인 리눅스를 기반으로 국내 표준 임베디드 리눅스 운영체제 및 통합 개발환경 도구 개발, 국책 연구 기관인 ETRI에서 Eclipse 기반의 Esto IDE를 개발하여 상용화함.

□ 표준화 주도를 통한 선진 기술 시장 진입
Qplus 커널 개발을 기반으로 운영체제와 IEEE 표준화 단체에 가입 및 주도

□ 핵심 인력 양성을 통한 기술 격차 해소
임베디드 시스템의 공개된 기술 스펙에 대해 정부 차원의 적극적인 핵심 인력 양성 정책

□ 기술 독과점 방지
특정 벤더 중심에서 공개SW 기반의 임베디드 운영체제 개발 기술과 다양한 제품군에의 적용은 기술 독과점을 방지함은 물론 다양한 기술 및 시장 진입이 가능해짐

아. Mobile Platform

(1) Mobile Platform 기술동향

과거 스마트폰은 PDA 폰이라는 이름의 PDA 기능과 폰의 기능이 물리적으로 합쳐진 단말에서 시작하였다. 반면 최근의 스마트폰은 폰 중심의 기능이 PDA 기능을 흡수한 양상을 보이는데, 이를 통해 두 기능의 유기적 동작이 가능해졌다.
모바일 플랫폼은 스마트폰뿐 아니라 기본 기능만 제공하는 폰(vanilla phone)부터 시작하여 카메라 폰 또는 MP3 폰과 같은 기능폰(feature phone)에도 탑재되는

소프트웨어로, 폰의 하드웨어적 성능을 상위 응용소프트웨어에서 사용할 수 있도록 하는 역할을 한다. 휴대 단말에 탑재된 운영체제와 플랫폼을 구분하기도 하나 범용 운영체제가 사용되면서 운영체제부터 애플리케이션 프레임워크까지의 모든 소프트웨어를 일컫는 말로 사용되고 있다. 한 예로, Qualcomm의 CDMA 기술을 사용하며 MSM 칩을 주로 사용하는 한국의 경우 동사의 REX를 운영체제로 사용하고, 이 위에 무선인터넷 서비스를 위한 WIPI 등을 플랫폼으로 탑재하여 사용했다. 모바일 플랫폼은 모바일 단말에 탑재되는 소프트웨어로 상위 응용프로그램이 다양한 서비스를 제공할 수 있도록 하는 데 도움을 주는 소프트웨어를 말한다.

Apple은 휴대 단말기인 iPhone을 2007년 1월 9일 발표했다. Google은 2008년 출시 계획을 가진 구글폰(gPhone)의 플랫폼인 Android의 SDK를 2007년 11월 12일 공개했다. Apple의 iPhone은 차별화된 UI를 통해 마니아층을 형성시켜 iPod에 이어 Apple의 휴대 단말의 성공을 이끌어 냈다. Google의 Android는 인터넷 서비스 업체인 Google이 만들어낸 개방형 플랫폼으로 제품 출시 전부터 SDK를 공개하여 많은 개발자를 확보해 나가고 있다. 기존의 플랫폼들은 범용 운영체제가 지원되기 전의 휴대폰부터 진화를 해나가 불필요한 기능을 가지고 있는 등의 단점이 있었으나, 이 두 플랫폼은 스마트폰에서 시작하여 이런 단점을 가지고 있지 않다. WIPI, BREW, J2ME, Symbian, Windows Mobile 등 기존의 모바일 플랫폼 영역에 두 플랫폼이 새로 등장하여 모바일 플랫폼 경쟁은 더욱 심화되었다.

(2) 상용 모바일 플랫폼과 공개 모바일 플랫폼의 비교

(가) 상용 모바일 플랫폼

1) Symbian

□ 개요

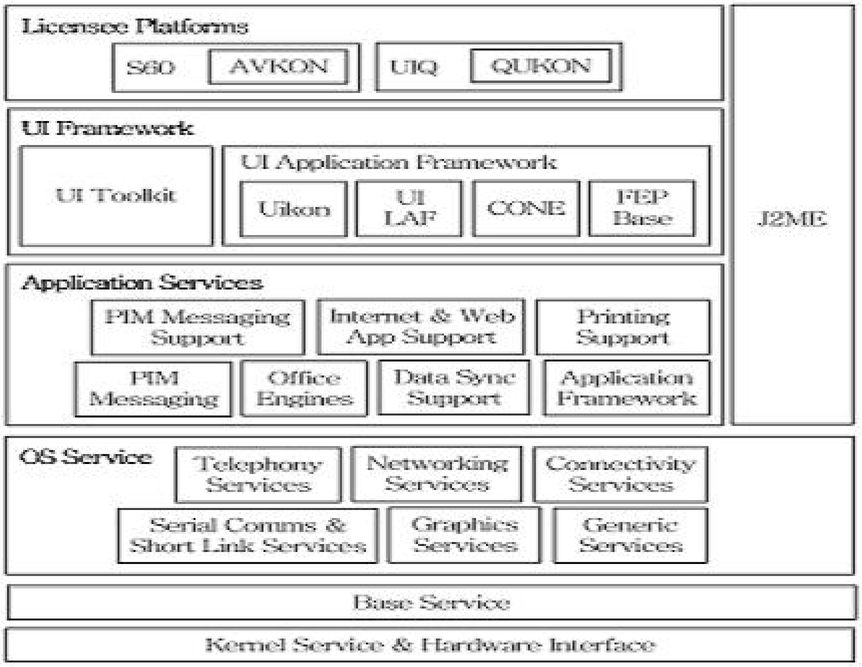
Symbian은 노키아 주도로 개발된 모바일전용 OS이다. 1998년 설립된 Symbian Alliance에서 Symbian의 개발 및 판매, 유지 보수를 담당하여 왔다. 지분 중 노키아가 56%를 소유하고 있어 사실상 Symbian의 방향을 좌우하여 왔다. ‘06년 기준 1억대의 휴대전화에 탑재되었으며, ‘08년 3사분기 기준 46.6%의 스마트폰 시장을 점유하고 있는 시장 점유율 1위의 플랫폼이다.

□ 특징

Symbian 운영체제는 기본적으로 마이크로 커널 기반의 경량 운영체제 구조를 갖추고 있고 네트워크 기능을 포함한 대부분의 기능이 마이크로 커널 상의 서버의 형태로 존재한다. Symbian 운영체제의 잘 정의된 OEM adaptation layer는 하드웨어 구

성 요소에 대한 지원 소프트웨어 및 통신 프로토콜 부분을 지원하기 위한 인터페이스 소프트웨어 부분 등이 서버 모듈의 형태로 정의되어 있어 다른 부분에 대한 고려 없이 비교적 독립적으로 새로운 하드웨어에 Symbian을 탑재할 수 있도록 돕는다. Symbian은 C++를 기본 언어로 하고 있으며 OPL, Python 등의 언어도 지원하며, 최근엔 J2ME도 지원한다. [그림4-4]는 체계적인 구조를 갖춘 Symbian의 구조도를 보여준다.

[그림4-4]에서 보는 바와 같이 Symbian은 S60과 UIQ의 플랫폼을 Symbian OS에 탑재한 구조를 가진다. Symbian OS는 OS service, application service, UI framework를 통해 다양한 기능을 제공한다. Symbian은 각종 표준 규격을 빠르고 완벽히 탑재하는 안정적인 플랫폼으로 Web SVG Tiny, AJAX, Flash Lite 2.0 등을 탑재한 브라우저를 통해 Web 2.0을 준비하고 있다.



[그림 4-4. Symbian 구조도]

□ 시장동향

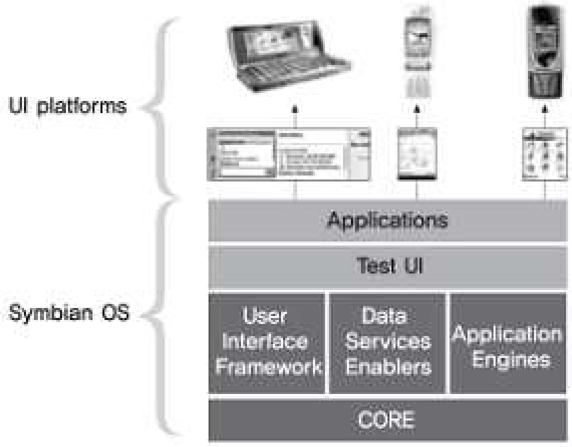
전 세계 단말기 1위 사업자인 노키아는 2009년 5월경 '오비 스토어(Ovi Store)'를 9개국 시장에 정식 출시하고, 게임, 비디오, 위젯, 팟캐스트, 개인화 콘텐츠, 위치 기반 애플리케이션 등을 제공할 계획이다. 노키아의 경우 '08년 2Q 기준 57.1%(Gartner)의 스마트폰 시장점유율로 대규모의 소비자 접점을 보유하고 있으며 자사 단말기에 탑재된 심비안(Symbian) 플랫폼의 확산은'오비 스토어'의 활성화의 Key Factor가 될 것으로 보인다. 특히 노키아는 심비안 OS를 오픈 소스화 함으로써 자사 단말뿐 만 아니라 다양한 단말의 플랫폼으로 작동함으로써'오비 스토어'활용 기반을 확대하고

자 하고 있다. 오비 스토어의 활성화는 결국 스마트폰 시장 및 플랫폼 시장에서의 시장 지배력을 유지시킬 수 있는 기반이 될 것이다. 뿐만 아니라 노키아는 Navteq 인수를 통해 Map을 확보함으로써 단말기의 GPS와 Map을 활용한 다양한 애플리케이션 개발의 토대를 마련해놓고 있다.

□ 기술동향

Symbian은 다양한 단말을 S40, S60, S80, S90으로 나누어, 각 군별로 기본 플랫폼을 제공, 빠른 단말 출시를 지원하고 있다. S40은 저 사양 단말을 위한 플랫폼이며, S60은 full-featured 스마트폰을 위한 일반적인 플랫폼으로, 현재 가장 많이 사용되고 있다. S80은 기업용, 전문가 수준의 단말을 위한 플랫폼으로 워드프로세서, 웹 브라우저 등 고급 소프트웨어를 탑재하고 있으며, 가장 높은 사양인 S90은 PDA, Netbook 등 고사양 단말을 위한 플랫폼이다.

Symbian은 Psion사의 EPOC을 기반으로 진화해왔으며 현재 Symbian OS 9.5가 발표되었다. [그림 4-5]는 Symbian 플랫폼의 구조도를 간략화 한 것이다. Symbian은GSM 및 CDMA, 3G, HSDPA 등의 기본 이동통신 및 WiFi 802.11,DVB-H, ISDB-T, OpenGL ES, 2M픽셀 카메라 등의 근거리 통신, 방송망, 그래픽 등을 제공한다.



[그림 4-5. Symbian 플랫폼의 간략 구조도]

노키아는 ’08년 6월에 Symbian의 모든 지분을 사들이고 공개 소프트웨어로 전환하겠다는 계획을 발표하였다. 공개되는 Symbian의 개발 및 유지보수를 위해 Symbian Foundation을 설립하고 ’09년 상반기 오픈 일정을 추진하고 있다. 이는 서비스 사업자로서의 영역 확장을 위하여 Symbian을 기본 플랫폼으로 확산시키고자 하는 노키아의 전략을 추측할 수 있다.

2) Microsoft Windows Mobile

□ 개요

MS의 Windows Mobile은 Pocket PC 2000 운영체제에서 시작하여 최근까지 여러 차례 업그레이드되어 왔으며, 가장 최근 버전은 윈도우 모바일 6.1로 '09년에 신규 버전이 출시될 예정이다. 이 플랫폼은 스마트폰 시장에서 2004년 23%에서 2008년 12%로 떨어졌으며, 현재 13%의 스마트폰 시장 점유율을 보이고 있다. PC사용자가 모바일에서도 Windows 플랫폼을 사용하도록 연계하는데 중점을 두고 있으며, 주요 업체의 견제를 피해 다수의 마이너 사업자를 통해 모바일 시장에 진입 하고 있다. Windows Mobile은 서버와 모바일 간 컴퓨팅 환경이 자동 동기화되고 연계되는 서비스를 통해 사용자 확보에 주력하고 있다.

□ 특징

Windows Mobile은 다양한 제품군으로 다양한 디바이스를 지원한다. 최근 발표된 Windows Mobile 6.0은 C# 기능을 강화하여 더 빠른 응용프로그램 개발을 가능하게 하였다. Windows Mobile계열은 C, C++, C#을 지원하며 데스크톱에 탑재되는 Windows와 상당부분 동일한 API를 제공하여 쉽고 빠른 개발 환경을 제공한다. Symbian과 마찬가지로 오랫동안 사용되어 안정성과 성능이 검증된 플랫폼으로 기본 라이브러리부터 UI까지 풍부한 기능을 제공하는 것이 특징이다.

Windows Mobile은 WindowCE 기반의 모바일 용 플랫폼으로 각 기능들이 모듈화 되어있어, 개발 하고자 하는 타깃의 용도에 따라 선택적인 기능 추가가 가능한 SW구조를 가지고 있다. 사용자에게 친숙한 데스크탑용 UI 및 MS오피스를 모바일에서 도 그대로 제공하며, 데스크탑과의 이메일 동기화 기능 등 모바일 사용자에게 데스크탑 응용의 연계성을 제공한다는 특징을 가지고 있다.

□ 시장동향

Microsoft는 Windows Mobile을 탑재한 디바이스는 누계로 약 5000만 대가 출하되고 있으며, 이 중 약 2000만 대는 2008년 중에 판매되었다고 말한다. PC에 비하면 아직도 적지만, 同사의 Windows Mobile 비즈니스가 급속히 확대되고 있는 것은 확실하다. Windows Mobile 6.5로 도입된 새로운 대기 화면과 Web 브라우저(Internet Explorer mobile)에는 휴대전화를 위한 한 손 조작을 의식한 업그레이드가 더해지고 있으며, 향후 Mobile이라는 말이 의미하는 디바이스는 “PDA”가 아닌, “Phone”이라는 것을 어필 하고 있는 것으로 보인다.

□ 기술동향

'09년 3분기에 예정인 마이크로소프트의'윈도 마켓플레이스(Windows Marketplace for Mobile)'의 경우 이미 유선 인터넷 환경에서 윈도우를 통한 플랫폼 장악력을 보유하고 있으며 다수의 강력한 윈도우 기반 애플리케이션 을 가지고 있어 기존 윈도우 기반의 개발자들을'윈도 마켓 플레이스'로 끌어들임으로써 무선 환경에서의 플랫폼 장악력을 이어가고자 하고 있다. 또한 마이크로소프트의 모바일 플랫폼인'윈도 모바일(Windows Mobile)'의 경우 기존 desktop 상의 윈도우와 상당부분 동일한 API를 제공하여 쉽고 빠른 개발 환경을 제공할 수 있으며 유무선 동일한 플랫폼 사용으로 유무선을 연동시킨 애플리케이션 개발 역시 용이할 수 있는 장점을 가지고 있다. 뿐만 아니라 마이크로 소프트 역시 유선 환경에서의 개발 노하우 및 SDK를 보유하고 있어 애플이나 구글 못지않은 편리한 개발 환경을 제공함으로써 개발자들을 유인할 수 있다.

3) BlackBerry(RIM)

□ 개요

Research In Motion(RIM)은 전 세계 무선 커뮤니케이션 시장을 대상으로 혁신적인 무선 솔루션을 설계, 제조, 마케팅 하는 선도적 업체이다. 다양한 무선 네트워크 표준을 지원하는 통합된 하드웨어와 소프트웨어 개발 및 서비스를 통해 RIM은 e-mail, 전화, SMS 메시지와 인터넷, 인트라넷 기반 어플리케이션 등 시간성이 강조되는 정보에 완벽한 접근성을 보장하는 플랫폼과 솔루션을 내놓고 있다. RIM 기술은 또한 다양한 제 3자 개발자들과 제조업체들이 무선 데이터 기능을 첨가하여 자사 제품과 서비스를 한층 개선하도록 한다. RIM의 수상 경력에 빛나는 제품과 서비스 및 임베디드(embedded) 기술로 이루어진 포트폴리오는 전 세계의 수천 개 고객사에서 이용되고 있으며 블랙베리 무선 플랫폼, RIM 무선 핸드헬드TM(RIM Wireless HandheldTM) 제품라인, 소프트웨어 개발 툴, 라디오 모뎀과 소프트웨어/하드웨어 라이선스 협약 등이 포함되어있다.

□ 특징

RIM은 포화된 미국시장에서 특정 그룹의 소비자들에게 특화된 제품을 만들어 공급한다. 예를 들어 지난 7월에 발표된 블랙베리 커브 제품은 소셜 미디어 마니아인 십대들을 위한 제품으로 사용자가 직접 찍은 사진인 동영상등을 유투브나 페이스북으로 업로드 할 수 있는 버튼을 갖추고 있다.

RIM이 휴대폰을 이메일과 접목해 휴대용 컴퓨팅 하드웨어 시장을 선도했지만 소프트웨어에 있어서는 선두 애플과 안드로이드 마켓으로 애플을 추격하고 있는 구글을 멀찍이서 좇아가고 있을 뿐이다. 앞으로 모바일 어플리케이션에서 어떤 모습을 보여주

느냐가 새로운 과제로 떠오르고 있다.

□ 시장동향

캐나다의 기업용 비즈니스 스마트폰 제조사인 RIM은 보안이 강화된 이메일 솔루션을 갖춘 기업용 스마트폰 'Blackberry'의 판매를 통해 북미 시장을 장악하였고, 유럽등지도 독자적인 시장을 구축하는데 성공하였다. 2008년 분기 스마트폰 시장점유율은 15.2%로서 Nokia 및 Apple에 이어 세계 3위를 기록하고 있다. 그러나 Nokia, Apple, 삼성전자 등이 기업용 스마트폰 시장에 진출하거나 진출을 고려함에 따라, 멀티미디어 및 GPS 등의 기능을 추가하여 개인용 소비자 시장 공략을 강화하고 있다. 또한 2009년 초에는 Apple의 App Store와 유사한 Application Marketplace인'Blackberry Application Store Front'와 On-device Application Center'라는 마켓플레이스를 출시할 예정이다. 5%, 기타가 9%를 차지하고 있다. 판매실적은 2006년 6.4백만 대에서 2007년 3.8백만 대로 116%가 증가하였으며, 2008년에는 19.4 백만 대가 판매될 것으로 전망되었다.

그러나 2008년 12월 초 발표된 분기 실적 전망에 따르면 당초 예상치보다 성장세가 둔화될 것으로 전망된다. 9월 마감된 회계연도 3분기 매출이 27억5000만~27억7000만 달러 수준으로 예상된다고 발표하였다. 2007년 기준 지역별 판매 비중을 보면 북미가 76%로 가장 크며, 유럽이 15%, 기타가 9%를 차지하고 있다. 판매실적은 2006년 6.4백만 대에서 2007년 13.8백만 대로 116%가 증가하였으며, 2008년에는 19.4 백만 대가 판매될 것으로 전망되었다.

그러나 2008년 12월 초 발표된 분기 실적 전망에 따르면 당초 예상치보다 성장세가 둔화될 것으로 전망된다. 9월 마감된 회계연도 3분기 매출이 27억5000만~27억7000만 달러 수준으로 예상된다고 발표하였다.

□ 기술동향

RIM과 HP는 전략적 제휴를 맺고 기업고객 시장을 겨냥해 블랙베리 플랫폼 기반 비즈니스 모바일 솔루션 제공에 나설 예정이다. 이번 제휴로 두 회사는 블랙베리용 'HP 클라우드 프린트'와 블랙베리엔터프라이즈 서버용 관리 솔루션인 'HP오퍼레이션매니저' 등을 선보이고 기업 고객의 서비스 수준 및 생산성 제고, 비용 절감 등을 적극 지원한다는 전략이다.

클라우드 프린트는 블랙베리 사용자가 사무실이나 집, 심지어 길거리에서도 e메일, 문서, 사진, 웹페이지 등을 인쇄할 수 있도록 한 웹기반 솔루션이다. 오퍼레이션매니저는 블랙베리 엔터프라이즈서버, 메일서버, 데이터베이스, 윈도서버 등 다양하게 구성된 정보 시스템 환경을 중앙 집중식의 단일화 된 콘솔로 관리 및 운영할 수 있도록 해 비용을 줄일 수 있도록 개발됐다.

또 개인 고객 시장을 겨냥한 신제품 개발도 이어진다. RIM은 개인 사용자 시장을 겨냥해 블랙베리 스톰의 차세대 버전 개발을 계획 중이며 이미 관련 로드맵까지 수립한 것으로 알려지고 있다.

4) 맥 OS

□ 개요

아이폰의 성공은 콘텐츠와 플랫폼, 그리고 단말로 이어지는 수직 결합된 비즈니스 모형에 기반 한다. 즉 차별화된 UI를 가진 단말기에 자사의 플랫폼인 Mac OS X를 탑재하고, 아이튠즈 스토어를 포함하는 다양한 분야의 풍부한 콘텐츠와 애플리케이션을 제공한다. 아이튠즈 스토어는 콘텐츠 서비스를 무기로 애플의 플랫폼 확장을 가능하게 하는 핵심자원이다. 현재 아이튠즈 스토어는 6백만 곡의 음악과 6만5천개가 넘는 팟케스트(미국), 만개를 넘는 뮤직비디오(미국), 350개 이상의 TV프로그램(미국), 2만 타이틀 이상의 오디오 북, 5백 개가 넘는 영화(미국)를 제공하고 있으며, 음악의 경우 2007년 7월말까지 25억 곡 이상이 팔렸다.

□ 특징

○ 혁신적인 UI과 Touch Tone방식 Access

애플은 차별화된 UI, 풍부한 콘텐츠와 애플리케이션과 함께 기존에 추구했던 폐쇄적인 비즈니스 모형을 지양하고, 아이폰을 위한 서비스 플랫폼을 개방함으로써 다양한 애플리케이션과 서비스들이 신속히 만들어지고 개발될 수 있는 환경을 제공한다. 이는 아이폰을 통해 제공되는 웹 애플리케이션에 제3의 개발자들의 참여를 허용함으로써 자신의 서비스 플랫폼을 중심으로 개방적인 생태계를 구현하겠다는 의미이다.

○ WiFi 지원으로 MNO의 망을 우회한 인터넷 접속 기능

애플은 또한 이동통신사업자의 영향력을 우회한 새로운 서비스 채널도 확보하고 있다. 애플은 아이폰 출시 3개월 후인 9월 28일 아이폰을 통해 인터넷 접속과 다양한 인터넷 기반 서비스를 지원하는 iPhones 업데이트를 제공하기 시작했다. 이로써 아이폰을 통해 이동통신사의 사설망을 통하지 않고도 WiFi라는 대체경로를 통해 인터넷 서비스를 이용할 수 있게 되었다. 또한 블루투스 기술을 채택해 단말간의 P2P 커뮤니케이션도 지원함으로써 단말기의 기능성을 극대화하고 있다.

○ 3자 개발 애플리케이션 수용을 통한 솔루션 확장 기능

이러한 애플의 노력과 아이폰의 기능은 이동통신사업자의 비즈니스 모델에 잠재적인, 그러나 치명적인 위협이 된다. 독점 계약자인 AT&T의 망을 회피하여 인터넷 서비

스를 제공할 수도 있고, 더욱이 모바일 VoIP를 통해 음성서비스의 제공도 가능하기 때문이다. 현재 AT&T는 웹서핑 자체는 허용하는 대신 모바일 VoIP와 P2P 트래픽을 차단함으로써 이러한 위험성을 차단하고 있지만, AT&T와의 독점계약이 끝나는 2009년부터 애플은 아이폰을 이용하여 독자적으로 모바일 VoIP를 제공할 수도 있다. 이러한 상황에서 애플이 보유한 강력한 서비스 플랫폼과 아이폰의 혁신적인 UI와 기능들은 이동통신 시장의 가치사슬을 주도하는 강력한 힘이 될 것으로 보인다.

그러나 구글의 안드로이드가 등장하고 노키아의 새로운 전략이 구체화되는 상황에서 애플의 아이폰 전략이 지속적으로 성공을 거둘 수 있을지는 예단하기 어렵다. 다양하지 않은 단말기 라인업과 특정 이동통신 사업자를 통한 제한된 서비스 제공은 네트워크 외부성 효과가 중요한 모바일 시장에서 고객의 선택과 서비스의 확장을 제한할 수 있을 것이기 때문이다. 그럼에도 불구하고, 애플과 아이폰은 이동통신사업자 중심의 폐쇄적인 시장구조에서 강력한 서비스와 혁신적인 단말기와의 결합을 통해 네트워크와 주파수라는 현실적인 한계를 극복하고, 새로운 모바일 시장에서 서비스의 플랫폼화를 통해 시장을 주도해 갈 수 있음을 실증적으로 보여준 사례다.

□ 시장동향

가장 먼저 애플리케이션 마켓을 오픈한 애플의 '앱스토어(App Store)'는 2008년 7월에 개시된 이래로 등록 어플리케이션 수만 3만 8천개에 이르며 현재 10억 건 이상의 다운로드 수를 기록하고 있고 1억 6,000만 달러 매출을 달성한 것으로 예상되고 있다. 애플의 경우 기존에 시장을 형성하고 있던 아이팟(iPod) 및 아이튠즈(iTunes)를 중심으로 충성(Loyalty) 높은 가입자 기반을 보유하고 있는 상황에서 이를 '앱스토어(App Store)'로 전이시킴으로써 다수의 개발자들을 앱스토어로 유인하고 있다. 이렇게 다양한 아이디어를 가진 개발자들의 애플리케이션 마켓 참여는 다시금 소비자들을 동 시장으로 유인함으로써 애플의 주력 제 품인 아이폰, 아이팟의 판매 확대로 연계시키는 선순환 구 조를 형성하고 있다. 뿐만 아니라 애플은 이미 통합된 형태 의 우수한 개발툴(SDK)을 보유하고 있으며 이를 개발자들에게 공개하고 매 버전별로 업그레이드하여 제공함으로써 개발자들이 쉽게 애플 플랫폼 위에서 개발할 수 있도록 지원하고 있다. 한편, 소비자에게는 Wi-Fi 및 PC-Sync 등 Side-loading을 제공함으로써 앱스토어에서 저비용으로 다양한 애플리케이션을 이용할 수 있는 유인책을 제공 하고 있다. 위와 같은 애플의 핵심역량은 애플의 앱스토어를 선호하는 개발자 및 이용자를 만들어냈으며 앱스토어 시장 확대에 기반이 되고 있다.

□ 기술동향

월스트리트저널 8. 11일자의 기사(iPhone Software Sales Take Off: Apple's Jobs)에 따르면 애플 앱스토어의 1달간 실적은 놀라움 그 자체라고 할 수 있다. 한 달간 총 6천

만회의 다운로드를 기록하였으며, 1일 평균 1백만 달러의 매출을 올려 총 3천만 달러의 매출을 기록한 것이다.
이전까지 모바일 SW들은 이동통신사의 플랫폼에 절대적으로 종속되어 있다. 망 개방 이전까지는 이동통신사의 플랫폼에 종속되어 중복 개발에 따른 부담감이 크게 작용했으며, 수익 배분에 있어서도 이동통신사의 입김과 "을"이라는 불리한 입장에 서 있었던 것이 사실이다.
애플 앱스토어의 등장은 그 동안 이동통신사 중심으로 형성되어 온 모바일 인터넷 시장에서 이동통신사의 영향력이 축소되는 결과를 가져올 것으로 예상된다. 이동통신사는 단순 망 제공자로 입지가 축소되고 특히, 단말기 및 OS/플랫폼 제공자와 SW 개발자들의 영향력이 확대될 것으로 전망된다.
앱스토어 비즈니스의 장점은 iPhone SDK를 제공하고 명확한 수익분배 모델(개발자:애플=70:30)을 제시함으로써 전 세계를 대상으로 수많은 개인 개발자와 개발 회사들을 우군으로 확보할 수 있다는 것이다. 이것은 SW의 영향력이 확대되고 있는 모바일 인터넷 시장에서 애플에게 가장 강력한 경쟁 무기가 될 것으로 전망된다.
또한 개인 개발자 및 개발 회사들도 이동통신사를 거치지 않고서도 전 세계적으로 광범위하게 직접 소비자들과 접촉하여 수익을 낼 수 있으며, 히트할 경우 막대한 수익을 얻을 수 있다는 점에서 반가운 일이 아닐 수 없다.

5) 팜(Palm)

□ 개요

Palm은 오래전 HW사업부인 Palm과 SW 사업부인 Palm Source로 분리를 하였다. Palm과 Palm Source는 상호 계약에 의해 파트너 관계를 계속 유지하면서 단말을 계속해서 개발해 왔으나 얼마 전(2007년으로 기억한다) 계약이 끝나면서 독자적인 길을 걷게 된다.
그 직전, Palm Source는 Access에 합병을 당하면서 이미 계열회사라는 개념이 없어져 버렸다. Access에 합병된 Palm Source는 Palm OS를 고도화만 해서는 최신 기술 트렌드를 따라가기에는 무리가 있다고 판단하고 Linux 기반의 ALP라는 플랫폼을 개발한다. ALP는 Linux 기반이지만 Application을 구동하고 관리하는 부분의 메커니즘은 Palm OS와 무척 닮아 있다. 게다가 Ghost라는 플랫폼 내의 에뮬레이터에 의해 고전 Palm의 어플리케이션과 Binary 호환이 된다.
이로 인해 Palm OS가 실질적인 upgrade가 중단이 되고, HW 업체인 Palm은 독자적인 플랫폼이 필요할 수밖에 없는 상황이 되었다. 그렇게 해서 개발에 들어간 자체 플랫폼이 Palm Pre OS이다.

□ 특징

Palm Pre에 채용된 Web OS의 주요한 특징 5개를 정리하자면 Cards, Gestures, Notifications, Universal Search, Multi Tasking을 이야기 할 수 있다. 브라우저를 떠난 Web Application이라는 새로운 발상을 실현에 옮긴 Palm은 단순한 Web 개발자 끌어안기 외에 5가지의 강력한 무기를 장착하여 시장에 승부수를 던졌다.

각 어플리케이션의 View Mode는 'Card' 단위로 관리하고 사용자들과는 Double Tap, Swipe up, Slow swipe up, Swipe left 등과 같은 다양한 Gesture를 통해 Communication 한다. 세계 최초의 진정한 Multi Tasking을 지원하는 모바일 OS라는 것도 사용 편의성을 높게 해준다.

□ 시장동향

Sony가 미국에서 휴대 정보 단말(PDA)의 개발 판매를 정지한다고 발표함에 따라 「Palm OS」를 공급하는 Palm Source(NASDAQ:PSRC)의 대기업 라이선스처는, 미국 및 유럽에서는 PalmOne (NASDAQ:PLMO) 뿐이 남지 않게 되었다. 덧붙여서 양사 모두, 구 Palm이 소프트웨어 부문과 하드웨어 부문에 신사 해 독립한 회사다. PalmOne은 IDC의 조사에 의하면, 시장점유율 감소가 계속되고 있지만, 탑 시장 점유율을 아직 견지하고 있다. PalmOne으로써는, Sony 제품의 유저로, palm OS 플랫폼에 집착하는 층을 손에 넣을 수 있겠지만, Microsoft의 「Windows Mobile」진영 각사, 즉 시장 점유율로 PalmOne에 압박하고 있는 Hewlett-Packard나, Sony와 거의 동 수준이었던 Dell, 그리고 상위를 넘보는 도시바 등이 palm OS에 구애됨을 가지지 않는 부동층을 손에 넣어서 PDA 시장에서 우위에 서는 절호의 기회라고 할 수 있다.

플랫폼 경쟁에서도 큰 영향이 나올 가능성이 있다. Gartner의 조사에 의하면, 2004년 제 1/4분기의 플랫폼별 시장점유율은, PalmOS가 40.7%, Windows CE가 40.2%라고 하는 상태로, 거의 어깨를 나란히 할 정도로 Windows 진영이 뒤쫓아 오면서 Palm OS진영은 후퇴가 계속되고 있다.

□ 기술동향

이미 알려진 바와 같이 Palm Pre는 Sprint향으로 첫 번째 단말을 개발 중이다. Sprint의 3G 망을 통해 서비스될 예정이며 음성, Data, SMS 등을 모두 포함하는 파격적인 요금제를 준비 중이다. 다양한 3rd Party Application도 준비 중인 것으로 알려져 있는데 지금 밝혀진 바로는 Navigation, Sprint NFL 등이 개발 중이다. iPhone의 App Store와 동일한 Market Place도 개발 중에 있으나 3D Game과 같은 화려한 Application은 Java Script의 기술적인 한계로 1차 버전에서는 제외될 것으로 알려져 있다. 또한 2009년 말까지는 Flash도 지원할 계획이 있음을 Palm에서 밝히고

있다. 최종 릴리즈는 2009년 중반이라고 이야기 되고 있으나 예정보다 조금 늦어질 것 같은 느낌이다.

6) BREW

□ 개요

BREW는 핸드셋용 응용프로그램 개발을 위한 표준 개발 환경을 제공한다. 핸드셋 칩 제조업체인 Qualcomm에서 개발한 플랫폼으로 빠른 개발 주기와 기능 개선과 확장이 빠르다는 장점을 가지고 있다. 하위 운영체제의 소프트웨어에 대한 지식 없이 BREW에서 제공하는 풍부한 API를 쉽게 응용프로그램이 가능하며, BREW 응용프로그램은 바이너리로 동작하므로 빠른 실행이 가능하다는 장점을 가지고 있다.

□ 특징

BREW는 핸드셋에서 데스크톱과 유사한 환경을 제공하여 사용과 개발이 용이하다는 장점을 가지고 있다. 다른 윈도 시스템들과 같이 이벤트에 기반을 둔 구조를 가지고 있고, C 언어를 통해 개발하며, 개발사 및 제조사에 의해 부족한 기능을 확장할 수 있는 등 풍부한 편의를 제공하고 있다[10]. Qualcomm의 uiOne은 개선된 UI 프레임워크로 커스터마이징과 퍼스널라이징이 가능한 프레임워크다. BREW의 extension으로 'Data is the UI' 개념을 사용하여 UI를 play 할 수 있는 데이터로 표현한다[11]. 이를 위해 TrigML을 UI를 표현하기 위한 마크업 언어로 사용하였다. TrigML은 Trigenix의 기술로 2004년 10월 Qualcomm에서 회사를 인수하면서 uiOne 기술로 포함되었다.

Qualcomm은 BREW 4 시리즈의 디자인을 공개하면서 BREW 4 시리즈가 컴포넌트에 기반을 둔 플랫폼임을 밝혔다. 핸드셋 소프트웨어 위에 컴포넌트 서비스 환경을 두었다. 컴포넌트 서비스 환경은 파일 시스템, 컴포넌트 매니저, 스레드 및 프로세스 처리, 메모리 관리 등과 같이 휴대 단말에서 공통으로 요구하는 컴포넌트를 기본적으로 탑재하고 기능 확장을 위해 컴포넌트를 추가하는 형식으로 플랫폼을 확장할 수 있는 구조를 제공한다. BREW 4 시리즈에 처음 도입된 컴포넌트 서비스를 통해 Qualcomm은 범용 운영체제 환경에 플랫폼을 제공하고자 한다.

(나) 공개 모바일 플랫폼

1) 모바일 리눅스 (Mobile Linux) (리모, LiMo)

□ 개요

삼성전자, 모토롤라, NEC, 파나소닉 등 4개 제조업체와 NTT Docomo, Vodafone 등 2개 통신사업자가 모여 '06년 초에 설립한 단체다. 현재는 유럽의 양대 이통사인

Vodafone과 Orange('08년 설립회원으로 합류)가 참여함으로서 상용폰을 위한 실질적인 리눅스 관련 표준화 단체로 인식되고 있다. LiMo의 설립 목적은 스마트폰에 필요한 임베디드 리눅스 기반 SW 플랫폼의 공동개발을 통해 개발 비용을 절감하고, Time-to-market 단축하는데 있다. LiMO의 플랫폼은 회원사에게만 공개된다.

LiMO에는 '08년 12월 기준 52개 회원사가 참여하고 있으며, 국내에서는 삼성전자, LG전자, SK Telecom이 이사회 멤버로, 이외에 KTF, 삼성SDS, 이노에이스, 아로마소프트 및 ETRI가 협력회원사로 가입한 상태이다. '08년 2월 Release 1 및 탑재단말 16 모델을 발표하였으며, 현재 Release 2를 개발 중에 있다.

LiMo Foundation은 표준규격이 발표되고 이를 수용하는 SW솔루션이 뒤따르는 일반적인 표준화 단체와 달리, 복수의 SW 솔루션 개발이 선행되고 있는 것으로 보이며, 그 중 이동통신사의 선택에 의해 자연 정리될 전망이다. GPL이 적용되는 리눅스OS 외에 UI 및 솔루션 등은 pluggable한 구조로 설계, 업체의 IPR이 유지될 수 있는 구조를 제공한다.

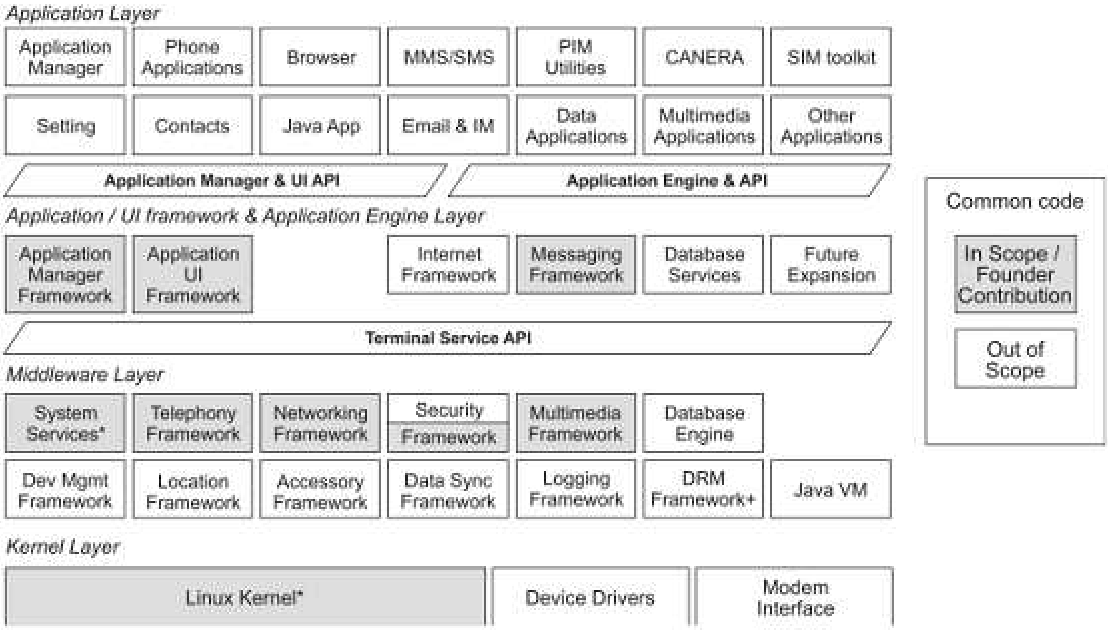
□ 특징

가능한 경우라면 LiMo 재단 플랫폼 아키텍처는 해당 플랫폼의 개발과 이용을 촉진하기 위해 기존 표준과 오픈 소스 프로젝트를 병행하여 활용한다. LiMo 재단은 투명하고 균형된 회원 기여 프로세스를 통해 커뮤니티 기반의 독점 개발이 지닌 이점을 활용함으로써 두 가지 접근 방식의 장점을 갖춘 플랫폼을 개발하고 있다.

LiMo 재단은 풍부한 기능성과 확장성(Scalability), 모바일 임베디드 시스템 내 우수한 성공사례, 소비자 전자제품이나 자동차 등 다른 분야와 쉽게 교차 플랫폼 화할 수 있는 적용 범위, 그리고 풍부한 리눅스 개발 인력을 모바일 분야에서 활용할 수 있는 잠재력 등 여러 가지 이점 때문에 리눅스를 LiMo 재단 플랫폼의 핵심 기술로 선택했다.

애플리케이션은 잘 정의된 API와 인터페이스 모델을 통해 미들웨어 컴포넌트를 액세스할 수 있다. 디자인 유연성을 위해 미들웨어 컴포넌트는 C나 C++로 구현될 수 있다. 애플리케이션 매니저 프레임워크와 애플리케이션 UI 프레임워크 등 2개 아키텍처 컴포넌트는 C 인터페이스와 C++ 인터페이스 둘 다 제공한다.

LiMo재단 플랫폼은 하드웨어 독립적으로 디자인되어 LiMo형 핸드셋을 만드는 LiMo재단 회원사가 어떠한 하드웨어 칩셋솔루션을 선택해도 무방한 유연성이 있다.



[그림4-6. LiMo 플랫폼의 전체 구조]

□ 시장동향

LiMo는 2007년 초 공식적인 출범을 하였으나, 실질적인 규격작업은 현재의 LiMo 창설멤버들을 중심으로 이미 이보다 3년여 전부터 진행되어 왔다. TLF가 리눅스의 공공성에 초점을 맞추고 있다면, LiMo는 리눅스의 모바일 사업성에 초점을 맞추고 있다고 할 수 있다. 따라서 모바일 리눅스 간의 상호환성 같은 공공의 이익보다는 단말기 개발의 생산성, 비용절감 등에 관심을 둔다. 또한 각 기업 제품의 특성을 침해하지 않기 위해 사용자 인터페이스 부분에 대한 규격은 최소화하였으며, 리눅스 운영체제 그 자체에 대한 규격을 중점적으로 규정하고 있다.

LiMo의 또 다른 특이한 점은 스스로 표준단체가 아님을 강조한다는 것이다. 기존 표준단체들이 표준규격을 정의하는데 그 목적을 두고 있는 반면, LiMo는 규격 및 결과물(바이너리 및 소스코드)을 동시에 출시하고 있다. 하지만 LiMo가 비공개 단체인 관계로 이러한 결과물들은 회원사에게만 공개되고 있다.

현재 LiMo의 멤버들은 창설멤버(Founder Level Member), 코어 멤버(Core Level Member), 참여멤버(Associate Member) 등 3단계로 구성되어 있다. 창설멤버는 모토로라, NEC, NTT도코모, 오렌지, 파나소닉, 삼성전자, 보다폰 등 7개 업체이며, 코어멤버에는 LG전자, SKT 등 9개 기업이 등록이 되어 있으며, 참여멤버로는 ETRI, KTF, 삼성 SDS 등 25개사가 참여 중이다.

□ 기술동향

LiMo는 올해 3월 스페인에서 있었던 Mobile World Congress에서 공식적으로 LiMo Platform Release 1 규격을 발표하였으며, 동시에 이를 탑재한 7개사의 18개 단말기를 시연하였다. LiMo Platform은 Java, Webkit, Native 형태로 구분되어 제공될 것이며, Release 1에 대한 이 세 종류 SDK들이 올해 중으로 공식 릴리즈가 예정되어 있다.

2) 안드로이드 (구글)

□ 개요

구글은 2007년 모바일용 플랫폼으로서 Android를 발표하였다. 그리고 향후 개발을 주도(Open Handset Alliance) 할 단체로 OHA를 설립했다. OHA에는 이동통신사, 칩셋 제조사, 핸드셋 제조사 들이 포함되어 있다. 2007년 11월에 초기 SDK 배포 후 수 차례의 수정을 거쳐 2008년 9월 Android SDK 1.0을 배포하였으며, 미국 T-Mobile을 통해 첫 Android 폰인 G1을 출시('08.10.22)하였다. G1은 대만의 HTC를 통해 제작되었으며, Kogan Technologies, Huawei Technologies, Lenovo, Sony Ericsson, Samsung 등도 Android 휴대폰을 출시할 예정이다. 2008년 9월 Strategy Analytics는 2008년 4분기 미국 스마트폰 시장의 4%(약 40만 대)를 Android 폰이 점유할 것으로 예측하였다. 또한, Google은 Android 폰용 응용을 다운로드할 수 있는 Android Market을 2008년 10월 22일 정식 오픈하였으며, 2009년 1사분기부터 유료 응용도 제공할 예정이다. 리눅스 커널 2.6을 기반으로 하위 라이브러리는 C/C++로 제작하고, 애플리케이션 프레임워크는 자바 실행 엔진인 Dalvik Virtual Machine으로 구성해 응용개발자에게는 자바 언어만 제공한다. 라이선스 및 특허 문제를 고려하여 SUN의 Java VM과 다른 Dalvik VM을 사용하고, 패키지 및 클래스도 JCP와 다른 Android만의 것으로 개발했다. Linux Kernel 기반으로 C/C++ 라이브러리 계층과 자바 VM이 제공되며, 그 상위에 애플리케이션 프레임워크가 구성된다. Linux Kernel 2.6 기반의 디바이스 드라이버, 보안, 네트워크 등의 핵심 리눅스 기능을 활용하며, 이메일 클라이언트, SMS, 브라우저와 같은 핵심 응용프로그램을 기본으로 제공하고, 기본 응용프로그램도 사용자에 의해 교체 가능해 많은 부분을 커스터마이징 할 수 있다. 데스크탑 PC와 동기화할 수 있는 기능은 의도적으로 지원하지 않는다. 유선에서 제공되는 구글의 웹 서비스를 휴대폰에서도 그대로 제공한다는 기본 전략 하에, 무선으로 Google 계정과 e-mail, calendar 등을 동기화하는 기능을 제공한다.

□ 특징

구글의 모바일 시장에 대한 도전은 2005년 안드로이드란 벤처기업을 인수하면서 시

작되었다고 볼 수 있다. 이후 단말기, 반도체, 소프트웨어, 이동통신업체들을 규합하여 개방형 모바일 기술 개발 연합체인 OHA(Open Handset Alliance)를 발족시키면서 모바일 시장 진출 전략을 구체화하기 시작했다.
이전까지 구글은 인터넷 검색과 광고 시장을 평정하였으며, 구글 앱스(Apps), 구글 앱 엔진(App Engine), 그리고 최근 출시된 웹브라우저 크롬(Chrome)으로 웹 애플리케이션 서비스 및 플랫폼 시장의 강자로 부상하였다.
이번에 공개된 구글 안드로이드는 웹 기반애플리케이션 서비스를 모바일 인터넷 시장으로 확대시킨 결정판이라 할 수 있다. 이로써 구글은 PC 플랫폼을 제외한 웹과 모바일 영역에서 애플리케이션 서비스를 제공할 수 있는 플랫폼을 모두 소유하게 되었다.
안드로이드 마켓의 SDK 1.0버전 공개, T-모바일의 안드로이드 탑재 스마트폰인 G1의 발표로 인해 애플리케이션 개발자 확보에 가속도가 붙을 전망이다. 구글 스마트폰이 많이 팔려야한다는 전제를 안고 있지만, 애플 앱스토어의 폐쇄성, 수익의 30%를 가져가는 수익배분 구조를 집중 공략하면서 애플에 대한 압박의 강도를 높여갈 것으로 예상된다.
구글은 안드로이드 마켓을 통해 개발자와의 직접적인 수익 배분보다는 인터넷 광고에서와 마찬가지로 모바일 인터넷에서도 검색과 광고를 통해 일차적인 수익을 얻을 것으로 전망된다. 또한 우리나라의 마스터CP와 같이 SW를 전문적으로 유통하는 회사를 유치할 가능성도 있는 것으로 예상된다.

□ 시장동향

애플의 앱스토어에 대응할 수 있는 수준의 서비스로 가장 먼저 공개한 것은 오픈소스 OS‘ Android’로 모바일 시장을 공략중인 구글(Google)이다. 구글은 ’08년 10월‘안드로이드마켓(Android Market)’을 오픈하였으며 ’09년 2월 유료화로 전환하여 애플리케이션 판매를 시작하였다. 구글의 등록 애플리케이션 수는 2,000여 개에 불과하나 구글이 유선 인터넷 상에서 보유하고 있던 핵심역량은 모바일 애플리케이션 마켓 활성화에 중요한 CSF로 작용할 것으로 보인다. 구글은 애플과 마찬가지로 이미 유선 인터넷 상에서 다수의 구글 마니아들을 형성하고 있다. 또한 구글의 대표적 웹애플리케이션인 구글맵의 존재는 모바일 애플리케이션 마켓에 진입하고자 하는 개발자들에게 다양한 활용 가능성을 기대할 수 있어 안드로이드 마켓으로의 참여를 이끌 수 있을 것이다. 뿐만 아니라 구글의 안드로이드 플랫폼은 자바 기반 개발언어를 활용하고 있어 기존 자바 개발자 풀을 확보할 수 있다는 장점을 가지고 있으며 안드로이드 플랫폼 이 제공하는 개발도구인 이클립스(Eclipse) 역시 편리한 개발환경을 제공하고 있어 다수의 개발자를 끌어 모을 수 있는 유인책이 될 것으로 보인다.

□ 기술동향

아이폰 발표 이후 많은 사람들이 구글 또한 모바일 폰을 출시할 것이라는 소문에 귀 기울이기 시작하였다. 사람들은 이를 "G-phone"이라 불렀으며, 그 출시일을 예측하곤 하였다. 하지만 세상에 G-phone은 출현하지 않았고, 그 대신에 수많은 G-phone을 만들 수 있는 Android라는 모바일 리눅스 기반의 SW플랫폼이 출시되었다.

Android는 OHA(Open Handset Alliance)라는 단체에 의해 출시되었다. OHA에는 구글을 중심으로 약 30여 개 기업들이 회원으로 등록되어 있다. OHA는 소프트웨어이의 공개성(Openness)을 중시하며, 공개성을 중심으로 모바일 기술의 기업측면에서의 생산성을 증진시킬 수 있다고 주장한다. 리눅스를 기반으로 하지만 소프트웨어의 공개성보다는 각 회원사들의 지적재산권(Intellectual property)을 중시하는 LiMo와는 달리, OHA는 모든 결과물을 소스코드 수준에서 일반에게 공개함 으로써 활성화를 도모한다. Android는 이러한 관점에서 OHA가 공개한 최초의 프로젝트 결과물이며, http://code.google.com/android/download.html에서 다운로드가 가능하다.

Android는 모바일 리눅스 기반의 SW플랫폼인 동시에 Dalvik이란 독자적인 자바 플랫폼을 지원한다. Dalvik은 clean-room JVM에 기반을 두어 기존 자바 기술 라이선스로부터 자유로워 많은 응용들이 Dalvik JVM기반으로 개발되리라 예상된다. Android는 또한 모바일 리눅스 기반이므로 C/C++기반 응용을 개발하는데 필요한 다양한 라이브러리 및 브라우저 엔진, DBMS 등을 제공한다. 이러한 라이브러리들을 기반으로 제공되는 프레임워크 미들웨어를 통해 모바일 단말 응용을 관리하고, 자원 및 사용자 인터페이스 등을 제공하기도 한다. 현재 Android에서 기본적으로 제공되는 응용은 이메일, SMS, 일정관리, 지도, 브라우저, 연락처 등이며, 이러한 응용들은 개발자의 선택에 따라 제거 및 새로 개발된 응용으로 대체가 가능하다.

Android가 적용된 단말기가 시장에 출시되고 있는 상황에서, 미국, 일본, 유럽 등지의 주요 이통사가 회원으로 가입되어 있는 상황이어서 그 가능성은 밝다고 할 수 있다. 회원의 등급은 없으며, 단지 회원사들의 업종에 따른 분류만이 존재한다. 현재 가입되어 있는 회원사들의 종류는 이통사, 단말기 제조사, 반도체 제조사, SW업체, 상용화 업체 등으로 구분되어 있다. 반도체 제조사의 경우 자사의 제품용 디바이스 드라이버 제공에 그 목표를 두고 있으며, SW업체의 경우 대부분 서비스 개발 업들로서, 자사의 서비스 응용을 Android 등에 탑재하는 것에 목적을 두며, 상용화 기업은 Android 자체의 상용화에 목적을 두고 있는 것으로 파악된다.

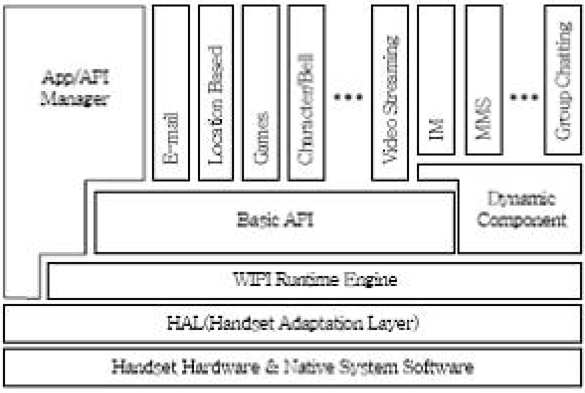
(3) Mobile 플랫폼 국내 개발 동향

(가) 국내 모바일 플랫폼

1) WIPI

□ 개요

WIPI는 무선인터넷 환경을 위한 플랫폼으로 과거 국내 이동통신사별로 GVM, SK-VM, MAP, BREW, CLDC/MIDP 등으로 분화되어 있던 모바일 플랫폼을 하나로 표준화한 무선인터넷 표준 플랫폼이다. WIPI는 이동통신 사업자, 단말기 제조사, 그리고 플랫폼 개발 업체들이 중심이 되어 만든 공개 표준이라는 특징을 가지고 있다. 소스 레벨의 호환성을 지향하는 WIPI는 플랫폼 표준화를 통하여 콘텐츠 제작 업체의 부담을 크게 줄였으며, 이동통신 사업자들도 다양한 콘텐츠를 빠르게 사용자에게 제공할 수 있게 되었다. WIPI는 2001년 9월 이동통신 3사의 공통 RFP 발표를 시작으로 2002년 3월에는 v1.0을 발표하였고, 2005년 8월 v2.0.2를 발표하였다. 현재는 범용 운영체제를 탑재한 휴대형 단말을 위해 WIPI 규격 3.0을 준비 중에 있다.



[그림 4-7. WIPI 플랫폼 구조도]

[그림4-7]은 현재 WIPI의 구조도이다. 핸드셋 소프트웨어 위에 HAL 계층이 있어 핸드셋 소프트웨어와 WIPI 플랫폼의 포팅을 도와준다. HAL 계층의 역할은 다양한 핸드셋 환경에서도 HAL이 요구하는 기본적인 API만 충족시키면 WIPI를 탑재할 수 있도록 하기 위함이다. WIPI의 핵심이라고 할 수 있는 WIPI Runtime Engine은 WIPI 콘텐츠를 동작시키는 주체다. J2ME의 경우 응용프로그램이 바이트코드로 컴파일 됨에 비해 WIPI 응용프로그램은 바이너리로 컴파일 되는데, WIPI Runtime Engine은 바이너리 형식의 WIPI 응용프로그램을 실행시키고 관리한다. WIPI 응용프로그램에 노출되는 Basic API는 C/Java API로 다양한 서비스가 가능하도록 풍부한 API를 제공한다. 2002년 8월부터는 Sun의 MIDP 프로파일을 Java API에 포함시켜 공인받은 API를 제공하는 기능적인 플랫폼이 되었다. 핸드셋의 발전으로 인하여 다양한 서비스가 필요하나 Basic API가 이를 지원하지 못하는 경우 dynamic component를 사용하여 WIPI의 기능을 확장하여 사용할 수 있다.

WIPI의 초기 탄생은 기존의 이동통신 단말기에서 시작하였다. 그러나, 텔레매틱스, DMB, RFID 등 다양한 분야에서 WIPI를 차용하여 사용하고자 하는 요구가 늘어 WIPI의 규격을 제안할 수 있는 열린 창구인 WSP(WIPI Standardization Process)를

제정하여 진행하고 있다. 현재는 WIPI3.0을 통해 다양한 단말과 범용 운영체제에 대응할 수 있는 규격을 준비 중에 있다.

(나) 표준화 동향

1) PG703 (WIPI기반 모바일 표준 플랫폼)

□ PG703의 구성 및 활동 범위

- 이동통신 3개사, 국책 연구소, 모바일 서비스 관련 솔루션 업체 및 기술 보유 업체들로 회원 구성
- 모바일 관련 서비스와 플랫폼 기술 및 표준화를 담당
- 국제 표준화 동향 탐색 및 참가
- 국내의 주요 이동 통신 관련 이슈 파악 및 솔루션 모색

□ 실무반

- AH7031 (단말 외부 인터페이스 특별반)
- WG7031 (OMA 실무반)
- WG7032 (모바일 표준 플랫폼 실무반)
- WG7033 (모바일 광고 기술 실무반)

□ 모바일 표준 플랫폼 (WIPI)

○ 주요 동향

▪ WIPI 3.0 규격 개발 중

- 다중 응용 실행 모델 강화 : 멀티쓰레딩, 동시성 제어, IPC 기능 보완
- 플랫폼 확장성 제고 : 모듈기반의 유연한 확장성 기능 확장
- UI 프레임워크 확장 지원
- 그 외 멀티미디어, I/O 등의 다양한 기능 확장 보완

▪ 표준화 진행 중인 WIPI 표준 제안(WSR) 규격

- P2P 서비스 위한 API (JXTA 규격을 근간)
- 모바일 데이터베이스 관리 및 접근을 위한 API
- DRM, M-payment, E-ticketing 등 모바일 전자상거래 지원을 위한 보안 API

○ 주요 이슈

▪ 각 이동통신사 별로 다른 API를 갖는 확장 기능에 대한 표준화

- 새로운 API 규격 제정은 낭비이므로 기존 API 활용위해 이동통신사간의 합의 노력 중

□ OMA

○ 주요 동향

▪ Open Mobile Internet 대비, 이동통신사업자의 신규 표준화 추진

- Phonebook, SNS를 결합한 통합 메시징

- Open Service Interface 개발(NGSI, CMI), GSMA와 협력

▪ 유무선기기의 통합 단말 관리 기술을 위한 공동 워크샵 수행('09.06월)

- OMA, UPnP, CableLabs, Broadband Forum, DMTF 참여

- 2차 CDM 회의 : 11월 Broadband Forum 회의

▪ WG 개편이 수행됨

- 기존 16개 WG이 11개로 개편

- 활동 효율화 및 향후 서비스별 Category로 결합

○ 이슈 및 정보

▪ GSMA RCS와 OMA 메시징 관련 표준의 개발 시기 Align 요구

▪ 2010년 11월 서울 회의 예정(Board, 기술총회, Working group 회의)

□ 범용 OS 기반 스마트폰

○ 주요 동향

▪ 스마트폰 기반 애플 개발

- WIPI가 미 탑재된 다양한 폰들이 속속 출시 예정

- WIPI가 담당했던 이동통신사간 공통 기능을 이통사별로 개별적으로 구현

- 콘텐츠 프로바이더들은 동일 OS에서도 별도의 구현이 필요함(해상도별, 제조사별)

▪ 다양한 기능, 고사양

- 2-3개의 네트워크 접속 기능 제공(3G, WiFi, Wibro 등)

- PC와의 다양한 sync, 호환성 제공 (Office 호환, 일정/메일 등)

- 특화 솔루션 탑재 가능(특정 고객 대상-법인 대상)

2) 표준화 전망

□ 표준화 추진 방향 (As-Is)

▪ 선진 주도의 표준화 기술

- 단말기 충전기 표준 인터페이스 (24핀, 20핀)

- 카메라 촬영음

- 단말에서 문자로 온 스팸을 바로 KISA로 신고(무료)

▪ 모바일 관련 표준화 활동 실적들

- 위피 중심의 모바일 플랫폼 구현

- OMA에 적극 참가하여 기고 및 동향 주도

▪ De Facto 표준들
- 이통사간의 각종 서비스 연동 규격들
- 규제 기관의 각종 고시와 권고 사항들

□ 표준화 추진 방향 (To-Be)
▪ 시장 파이를 확대하는 표준화
- 시장 형성 단계에서 표준화로 시장 활성화 유도
- Silotlr의 시장 구조를 개선할 수 있는 표준화
- 융합에 대비한 다양한 분야와 연계 및 연합
▪ 표준화 결과를 적극적으로 확대
- OMA에 적극 참가하여 기고 및 동향 주도
- 국제기관(GSMA 등)에 적극 참여
▪ 지배적인 이해 관계자를 적극 참여 유도
- 이통 3사들의 적극적인 표준화 참여
- 규제 기관의 적극적인 표준화 참여
- 제조사, 국책 기관의 적극적인 표준화 참여

(4) Mobile 플랫폼 기술 효과성

□ 기술 자립 기반 확보
- Open Mobile, Open API 주도의 시장에서 독자적인 모바일 플랫폼인 WIPI 개발 및 공개
□ 표준화 주도를 통한 선진 기술 시장 진입
- OMA를 비롯한 적극적인 표준화 참여 및 동향 주도를 통한 선진 기술 선점 및 모바일 세계 표준화에 국책 연구기관인 ETRI, 삼성, LG 등이 참여
□ 핵심 인력 양성을 통한 기술 격차 해소
- 공개된 기술 스펙에 대해 정부 차원의 적극적인 핵심 인력 양성 정책
□ 공개SW 기술 도입을 통한 수입 대체 효과 및 소프트웨어 시장 확장
- 국내 소프트웨어 업체의 기업 및 공공 소프트웨어 시장 진입
- 소프트웨어 기업 및 SI 업체의 해외 진출
□ 기술 독과점 방지
- MS를 비롯하여 특정 벤더에 종속적인 모바일 OS 및 모바일 플랫폼 기술 분야에서 공개SW 기반의 모바일 플랫폼의 등장으로 인해 기술 독과점 방지

5. 맺음말

공개SW는 개발 주기의 단축, 기술적인 호환성, 신뢰성과 안정성 등의 기술적 장점에도 불구하고 시장에서의 인지도는 아직 미약한 편이다. 이에 본 보고서는 SW 분야별 대표적인 공개SW와 상용SW를 비교 분석하고, 공개SW가 국내 기술 개발 동향에 어떤 영향을 주었으며, 각 소프트웨어 분야별 기술적 효과를 5개의 영역(빠르고 유연한 개발, 기술 자립 기반 확보, 표준화 주도를 통한 선진 기술 시장 진입, 핵심 인력 양성을 통한 기술 격차 해소, 기술 독과점 방지)으로 나누어 포괄적인 분석을 통해 국내의 공개SW 기반 기술적 효과에 대한 정보를 제공하고자 하였다.

SW 산업 분야에서 소프트웨어 공학의 발전과 컴포넌트 기반의 소프트웨어 개발 환경의 변화는 웹 2.0의 진화를 가속화하면서 IT 서비스의 "서비스로서의 소프트웨어(SaaS)“ 방향을 제시하게 되었고, 공개SW의 새로운 비즈니스 모델인 상용 공개SW의 등장으로 이어지게 되었다.

본 보고서는 운영체제(OS), Web 서버와 브라우저, WAS, DBMS, 통합개발환경(IDE), 임베디드 시스템, 모바일 플랫폼 분야를 공개SW의 기본 철학인 개방(Open)과 정보 공유의 개념을 기반으로 기술적 효과를 기술하고 있다. 기술적 효과로는 첫째, 코드의 재사용과 핵심 기술 및 정보의 공유, SW 패러다임의 변화를 통해 빠르고 유연한 개발이 가능해지며 이로 인해 소프트웨어 개발 주기의 단축이 가능하다는 것이다. 둘째, 공개SW의 기본 철학인 정보와 기술의 공유는 커뮤니티의 발전을 가속화하고, 접근과 개발이 어려웠던 분야의 다양한 제품의 양산으로 이어지며 기술 자립도를 확보하게 되었다. 셋째, 국내 대표 연구 기관인 ETRI, 대표적인 기업인 삼성과 LG, 기타 공공기관 및 민간에서 세계적인 표준화 기구에 가입하여 주도적인 역할을 수행하며 선진 기술 선점 및 국내 기술의 세계 표준화 기술 동향을 주도하고 있다. 넷째, 정부와 공공, 민간에서 다양한 분야에 정책적으로 소프트웨어 인력 양성을 위한 다양한 프로그램을 개발, 보급, 확대하고 있으며, 각 소프트웨어 분야에서 핵심 인력 양성이 배출되면서 선진국과의 기술 격차가 해소되고 있다. 다섯째, 공개SW 기반의 기술 개발은 국산 소프트웨어 개발의 견인차 역할을 하게 되며, 이로 인해 특정 벤더에의 의존성에서 점차 탈피하고, 기술 독과점 방지의 효과를 가진다.
공개SW는 국내 기술 개발 환경의 점진적인 개선을 주도하고 있으며, 선진 기술 습득의 기회를 제공한다. 기반 기술이자 핵심 기술인 OS, DBMS, 임베디드 시스템, 모바일 플랫폼 등 분야에서 비약적인 발전이 이루어지고 있으며, 공개SW 기반의 개발환경이 정부와 민간에서 충분한 이슈가 되고 있다.

개방(Open)과 정보 공유의 공개SW 철학이 새로운 IT 패러다임으로 변화하고 있으며, 국내 IT 기술 개발도 새로운 패러다임에 동참해서 정보 기술 선진국으로 진입하는 계기가 되어야 하며, 국내 공개SW 생태계의 선순환 구조를 형성하는 계기가 되었으면 한다.

참고문헌

1. David A. Wheeler 의 『Why Open Source Software / Free Software(OSS/FS)? Look at the Numbers! 』2003
2. 한국 소프트웨어 생태계와 대책 2008년 12월 김진형
3. OSSI(Open Source Software Institute)의 입장 2004년 장승권
4. 임베디드 소프트웨어 개발도구 기술 조주현 (주)코스모 대표이사
5. 오픈소스 소프트웨어의 가치와 경제적 영향에 대한 연구, 2007 송주현
6. 정보통신사업진흥원, "SW 산업동향-DBMS 시장동향 분석", 2008.06.23
7. 오픈 소스 소프트웨어와 개발도상국의 소프트웨어 산업 전략 2005년 6월 기술혁신 학회지 장승권, 고경민, 이회진
8. 공동자원의 커뮤니티 2007 이상묵
9. The Economics of Technology sharing Open Source and Beyond, 2006, Josh Lerner
10. 해외 SW시장 심층분석 2006 KIPA
11. 공개 소프트웨어의 이해 및 교육적 시사점 2006 한국교육학술진흥원 김현진
12. 오픈소스소프트웨어의 가치창출과 비즈니스모델 2005 서재고, 장승권, 박병수
13. 오픈소스 소프트웨어 연구 보고서 2002 한국소프트웨어진흥원
14. 공개 소프트웨어 도입 확대를 위한 정책방향 2007 한국지역정보개발원
15. 한국형 오픈소스 소프트웨어기술개발 전략 송 위 진 과학기술정책연구원 2002
16. oss for business/CSC Consulting Group
17. SW 지적재산권 동향『SW IPReport 제32호』2007. 11. 28 발행 5
18. towards Open Source Software adoption and dissemination Economical and Social Benefits of F/OSS Report D09 Version 2.7 12 July 2006 www.tossad.org
19. 미래의 소프트웨어 공학 기술 2006년 12월 정보과학지 고려대학교 김상수, 인호, 서울시립대학교 이병정, 서강대학교 박수용
20. 글로벌 모바일 단말 소프트웨어 플랫폼 동향 ETRI 전자통신동향분석 2008년 2월 윤민홍, 김선자

21. 공개SW 중장기 기반기술 기획 연구 2003년 12월 한국소프트웨어진흥원

22. 웹 2.0 기술 현황 및 전망 ETRI 전자통신동향분석 2006년 10월

23. 임베디드 운영체제 기술동향 ETRI 김홍남 임베디드 소프트웨어 연구단

24. 모바일 플랫폼 표준화 동향 및 향후 전망 KT 무선연구소 PG703 의장 허태범 2009년 8월 20일