

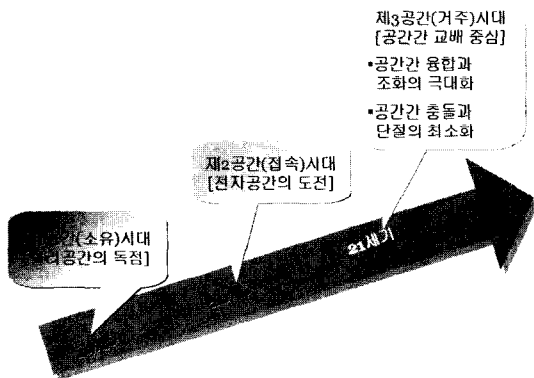
유비쿼터스 컴퓨팅 미들웨어 기술

윤 희 용

성균관대학교

I. 서 론

미국의 문명 비평가이며 미래학자인 앨빈 토플러(Alvin Toffler)가 제창한 제3의 물결 이후로 우리는 새로운 시대를 위한 준비를 하고 있다. 이는 <그림 1>과 같이 단순히 정보화의 혁명이라 하는 IT의 중요성뿐만 아니라 IT(Information Technology), BT(Bio Technology), NT(Nano Technology), CT(Culture Technology), ST(Space Technology)가 하나로 융합되어 자가 지식 성장 및 추론 가능한 환경 친화적 제3의 공간, 다시 말하면 물리적인 공간과 전자 공간을 하나로 융합한 시대를 만들기 위한 유비쿼터스 컴퓨팅 시대가 도래하고 있는 것이다. 그러나, 다른 한편으로는 대다수의 사람들이



<그림 1> 시대별 개발기술 패러다임 동향

본 연구는 학술진흥재단(KRF-2002-041-D00421)과 BK21 사업에 의해 지원되었음.

IT의 진보에 익숙해지기도 전에 벌써 미래의 새로운 기술로 유비쿼터스 컴퓨팅을 이야기하고 있어 많은 혼란을 가져올 수도 있다. 본 고에서는 유비쿼터스 컴퓨팅에 대한 간략한 설명과 유사 컴퓨팅 패러다임에 대해 알아본 다음, 유비쿼터스 컴퓨팅을 위한 미들웨어 기술 및 현재 유비쿼터스 컴퓨팅 미들웨어가 가지고 있는 문제점, 그리고 연구개발 동향에 대해 논하고 미들웨어 개발에 관한 향후 과제에 대하여 기술하고자 한다.

II. 유비쿼터스 컴퓨팅 및 관련 패러다임

유비쿼터스(Ubiquitous)란 영어 단어 자체의 어원을 보면 「도처에 널려있다」, 「언제 어디서나 동시에 존재한다」를 의미하는 것으로 다양한 컴퓨터가 현실세계의 디바이스, 환경 및 사물들 속으로 스며들어 언제, 어디서나, 어떠한 기기로도 통신 서비스를 이용할 수 있는 인간, 사물, 공간간의 최적의 컴퓨팅 및 네트워킹 환경을 구축함으로써 생활 속에서 자연스럽게 편리하게 컴퓨터를 사용할 수 있음을 의미하는 것으로, 1988년에 미국의 마크 와이저에 의해 제창되었다^[1].

유비쿼터스 컴퓨팅 환경을 구축하기 위해서는 극소형의 칩이나 센서를 도처에 우리들이 흔히 접할 수 있는 사물 속에 집어넣거나 몸에 지닐 수 있으며 이러한 디바이스들이 단거리 무선 통신을 통하여 언제, 어디서나 정보를 주고 받을 수 있어야 한다. 그러나, 한편으로는 인간 편의성의 극대화를 이룬 유비쿼터스 환경에서 개인의 사생활

활 침해와 보안이 심각한 문제로 대두되고 있다. 이는 개인의 위치와 행위가 모두 컴퓨터에 의해 파악되며, 모든 인간생활이 컴퓨터에 의존하는 시대가 되기 때문이다. 이러한 부작용을 가지고 있음에도 불구하고 유비쿼터스 컴퓨팅이 필연적인 진화 방향이라는 견해에는 아무도 이의를 달지 않는다. 이와 같은 내재하는 문제점을 해결함과 동시에 유비쿼터스 컴퓨팅의 성능을 최적화하기 위해서 효과적인 서비스들을 하드웨어 디바이스 벤더, 소프트웨어 또는 서비스 개발 업체에 제공하여야 한다. 이를 위해 이기종 하드웨어 및 운영체제간에 서비스 제공의 밀반침이 되는 미들웨어의 개발이 필수적이라고 말할 수 있다.

유비쿼터스 컴퓨팅 미들웨어는 사용자에 따른 서비스의 구별 및 에이전트의 학습에 따라 차별화된 서비스를 제공한다. 또한, 불안정한 네트워크 하에서도 안정적인 서비스를 제공받을 수 있도록 하는 네트워크 기술, 디바이스 간의 상태 정보를 수집하고 제어하는 서비스/자원 디스커버리 기술 및 사용자와 유비쿼터스 에이전트를 보호하고 인증 할 수 있는 보안기술 등이 포함되어 안정된 유비쿼터스 환경을 제공할 수 있도록 한다.

유비쿼터스 컴퓨팅 패러다임은 한 순간에 나온 것이 아니라 인간 편의성을 위한 여러 개념들을 통합하는 것으로 증강 현실(Augmented Reality), Nomadic 컴퓨팅, 착용 컴퓨팅(Wearable Computing), Pervasive 컴퓨팅 같은 유사한 패러다임들이 있다.

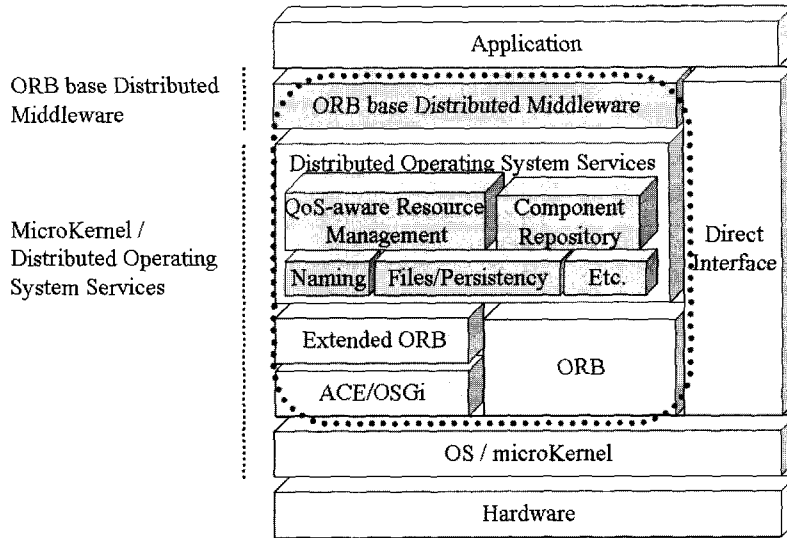
증강 현실(Augmented Reality)은 우리가 보고 있는 현실의 세계와 부가적인 정보를 가지고 있는 가상의 세계를 합쳐서 하나로 보여주는 것으로 가상 현실과 같이 현실 세계를 가상의 세계로 대처하는 것이 아니라 현실 세계를 보충해주는 것이다. 그 대표적인 예로는 GPS와 같이 사용자의 편의를 위한 가상의 세계를 도식화해주는 교통 정보 시스템을 들 수 있다. 이는 "Sentient Computing"이라 불리우기도 한다. Nomadic 컴퓨팅이란 [유목의, 방랑의]라는 뜻과 같이 장소나 디바이스들에 독립적으로 자신만의 정보환경을 구축하여 일관된 방식의 정보를

제공받을 수 있는 환경이다¹². 착용 컴퓨팅(Wearable Computing)은 현재 핸드폰과 PDA와 같이 소지한다는 개념보다는 의복과 같이 착용하고 다니는 환경을 말한다. Pervasive 컴퓨팅은 유비쿼터스 컴퓨팅과 매우 유사한 개념으로 사용자가 인식하기 전에 미리 사용자의 요구사항을 알아서 컴퓨터 스스로 사용자의 편의를 위한 최적의 환경을 구축하는 것을 말한다. IBM은 Pervasive 컴퓨팅을 현재 네트워크 환경으로 실현되고 있는 e비즈니스 환경을 자연스럽게 확장시킨 개념이라고 정의한다.

III. 유비쿼터스 컴퓨팅 미들웨어

미들웨어란 <그림 2>와 같이 운영체제와 응용 프로그램 사이에 존재하는 소프트웨어 계층으로 사용자에게 하부의 하드웨어, 운영체제, 네트워크에 상관없이 분산 컴퓨팅 환경, 원격 프로시저 콜, ORB, 메시징과 같은 서비스를 제공할 수 있도록 도와주는 소프트웨어로 정의할 수 있다¹³.

인간 친화적인 유비쿼터스 컴퓨팅 환경을 구현하기 위해서는 이기종간 다양한 유무선 통신프로토콜을 지원하는 네트워킹 기술, 내장형 프로세서 기술, SoC(System on Chip)기술을 기반으로 하는 능동적 센서와 이를 연결하는 센서 네트워크 기술이 필요하다. 또한, 이기종 환경에서 작동하는 다양한 센서, 디지털 디바이스 기기와 모바일, 그리고 유무선 컴퓨팅과 같은 환경에서 제공되고 실시간으로 수집되는 데이터를 사용자 친화적이고 언제, 어디서, 누구에게나 실시간으로 원하는 정보와 서비스를 예약하고 제공하는 기술이 요구된다. 그리고, 수집된 자료와 사용자의 자료를 취합하여 서비스를 결정하는 자가 지식성장의 추론기능과 이러한 서비스들이 환경 종속적이지 않고 응용프로그램간의 상호 운용성을 보장하는 사용자 투명성이 제공되어야 한다. 더불어 생활환경 곳곳에 산재해 있는 컴퓨팅 자원을 동적이고 효율적으로 사용하며 시스템에서 능동적으



〈그림 2〉 분산 운영체제와 미들웨어

로 적응하며 동작하는 Self-Configuration, Self-Organization이 가능하도록 해야 한다.

유비쿼터스 컴퓨팅은 분산되어 방대한 시스템 규모를 형성하고 있는 사물과 사람의 위치, 공간, 속성 정보들의 상황인식(Context Awareness)을 통하여 유기적인 통합 서비스를 제공하는 것을 목적으로 한다. 이는 인간 중심의 환경 인지적인 컴퓨팅 모델이기 때문에 각종 개인 지원기기(예, 스마트폰, PDA 등)뿐만 아니라 반도체 칩을 내장하고 있는 가전 기기들과 같이 인간의 모든 생활 환경에서 사용되는 디바이스를 포함하고 있다. 이러한 디바이스들은 서로 유기적인 통신을 이루기 위해 현재 널리 사용되어지는 유선 통신 뿐만 아니라 언제, 어디서라는 목적에 부합될 수 있기 위한 무선, 광역 모바일 네트워크, 센서 네트워크, Ad-hoc 네트워킹을 지원해야 한다. 다시 말하면 서로 다른 네트워크 환경에서도 다양한 장치들 간의 투명성을 제공해야만 하는 것이다.

이와 같이 다양한 하부 플랫폼과 상관없이 동작할 수 있는 유비쿼터스 컴퓨팅 미들웨어는 운영체제, 미들웨어, 응용프로그램 수준에 이르는 모든 영역에서 재구성, 재사용성, 적응성을 효과적으로 지원할 수 있는 컴포넌트 기반 소프트웨어

어로 개발되어 새로운 디바이스, 서비스 추가 및 변경에 대해 유연하게 동작할 수 있어야 한다.

광범위한 기술 스펙트럼과 각기 다른 하드웨어 디바이스, 네트워크, 운영체제, 입출력 장치 등이 혼재된 유비쿼터스 환경이 제대로 동작하기 위하여는 이기종 하드웨어 및 소프트웨어에 종속되지 않고 손쉽게, 자유롭게 외부 환경과 이음새 없는(seamless) 서비스를 지원할 수 있는 소프트웨어 플랫폼과 이기종 기기 및 서비스간의 상호 호환성, 사용자 투명성을 제공하며 사용자 및 시스템 보호를 위한 미들웨어의 중요성이 크게 부각되고 있다.

IV. 유비쿼터스 컴퓨팅 미들웨어 발전 동향

현재 선진국에서 연구 중인 유비쿼터스 미들웨어 관련 프로젝트들을 보면 <표 1>과 같은데, 미들웨어 연구를 좀더 세분하여 보면 센서 미들웨어 연구, 임베디드 S/W 및 미들웨어, 그리드 미들웨어, OSGI 표준과 Home Networking 미들웨어, 차세대 미들웨어, J2M 플랫폼과 JVM, KVM에 관한 연구 등으로 나눌 수 있다.

〈표 1〉 최신 유비쿼터스 컴퓨팅 미들웨어 관련 프로젝트¹⁴⁻⁶⁾

Project	특 징
CALAIS	<ul style="list-style-type: none"> • Context management에 관한 연구 • Context와 location awareness에 관한 연구 • 이기종 장치간의 통일된 인터페이스 제공 (event 기반) • 코바를 기반으로 한 4계층 layer 디자인 • 비교적 거대한 미들웨어
Context Fabric	<ul style="list-style-type: none"> • University of California at Berkeley 연구 프로젝트 • Context awareness 서비스 지원을 위한 하부 구조 연구 • 서비스와 application에 따른 sensor 제어
Ninja	<ul style="list-style-type: none"> • Context management에 관한 연구 • 컴포넌트 기반의 네트워크 인프라 구축 • Java Based 개발 환경 • 언어에 비독립적
Gaia	<ul style="list-style-type: none"> • 애플리케이션 실행을 위한 오디오 커맨드 언어지원 • 자원과 디바이스 서비스를 위한 각 서비스 정보를 저장 공간에서 관리 • 단일화된 객체인 BUS로 각 컴포넌트를 운영함 • 보안성을 고려한 설계 • 프로토콜에 비독립적
Oxygen	<ul style="list-style-type: none"> • MIT 대학 연구 프로젝트 • 다양한 언어 시스템 지원 • 리소스 디스커버리를 위한 INS(Intentional Naming System) 제공 • 이동성을 고려한 네트워크 인프라에 관한 연구 • 휴대용 장치 (handle device)들로 공간 네비게이션 기능 제공

1. 센서 미들웨어

유비쿼터스 컴퓨팅 환경에서는 PDA, 핸드폰과 같은 소형 디바이스뿐 아니라 생활 환경에 산재되어있는 센서 등으로부터 데이터를 수집하고 분산 처리한다. 이와 같은 센서를 포함한 이기종 하드웨어 및 플랫폼을 통합하여 데이터를 원활하게 연동할 수 있도록 지원하는 센서 미들웨어 기술이 요구된다. 이러한 센서 미들웨어 기술이 정적인 환경 하에 사용되는 홈 네트워크 시스템에 적용되고 있는 추세이다. 그러나, 실시간으로 변화하는 유비쿼터스 환경을 원활히 지원하기 위해서는 실시간 상황인지 및 위치인식 기술과 Nomadic 환경을 지원하기 위한 네트워크 기술, 그리고 마이크로 하드웨어의 특유성으로 인해 발생하는 저전력 및 작은 메모리 용량, 그리고 낮은 프로세싱 파워와 같은 특징을 수용할 수 있는 기술들이 요구된다. 이에 따라 기존의 미들웨어 기술과는 다

르게 하드웨어의 특성을 고려한 마이크로 운영체제 기술, 무선 센서 네트워크와 원격제어 및 센서 분산 시스템에 관한 연구가 수행되고 있다. 이러한 연구를 통하여 데이터 수집, 서비스 제공 등이 시스템 또는 환경에 대한 제한 없이 확장 가능할 수 있도록 한다.

이러한 센서 미들웨어로 유럽에서는 “사라지는 컴퓨팅 계획 (Disappearing Computing Initiative)”을 중심으로 잉크에 센서를 넣어 종이의 활용도를 높이는 Paper++과 같은 유비쿼터스 환경을 위한 프로젝트를 진행 중이며, 미국에서는 UCLA의 “Sensor-Ware”를 통하여 센서 네트워크의 유동성과 동적인 데이터를 지원할 수 있도록 하는 센서 미들웨어, 그리고 국내에서는 KAIST의 “MICROS (Micro Information and Communication Remote Object-oriented Systems)” 프로젝트를 통해 센서 무선 네트워

크 기술 및 데이터 수집/관리를 통해 서비스를 제공하는 유비쿼터스 도시를 위한 연구가 수행되고 있다.

2. 임베디드 S/W 및 미들웨어

기존의 임베디드 소프트웨어 및 임베디드 OS는 특정의 하드웨어나 OS만을 지원하기 위해 개발되었으며, 이에 더하여 특정 기능 또는 서비스만을 위해 개발된 통합 개발 환경이 대부분을 차지했다. 현재 미국이나 영국 및 세계 각국에서 연구 중인 임베디드 시스템 연구는 특정 임베디드 OS만을 지원하는 관계적인 통합개발 환경이 대부분으로, 임베디드 환경에 주로 사용되는 운영체제로 pSOS, VxWorks, Embedded Linux, WinCE, Palm 등이 있다. 이러한 OS들을 위한 통합 개발환경은 대부분 지원되지 않고 있는 실정이며 Palm, OSEKturbo, Linux, QNX와 같은 일부 운영체제들은 각기 회사에서 제공하는 SDK를 사용해 CodeWarrior IDE 환경하에 C/C++/Java로 개발 및 디버깅, 그리고 사용자 인터페이스를 디자인 할 수 있도록 통합 개발 환경을 지원한다. 이 외에 Borland, IBM, HP, INTEL 등과 국내 연구기관인 ETRI가 참여하고 있는 Eclipse Platform은 Embedded Java Program, C++ Program, Enterprise JavaBeans 및 웹 사이트 개발을 지원한다. Eclipse의 가장 큰 장점은 Plug-In 개념을 적용하여 XML과 Plug-In API를 사용한 틀의 확장이 용이하도록 했다는 점이다^[7].

마지막으로 MS사는 기존의 Visual Studio에 Net 프레임 워크와 Embedded Visual C++을 추가해 통합 개발환경인 Visual Studio.Net을 발표였다. 그러나, 단지 MS 운영체제 (Windows, WinCE)만을 지원하는 단점이 있다. 이 외에 모바일 임베디드 시스템 하드웨어의 특성에 맞도록 운영체제에서 지원하는 서비스 재구성 기능이 있는 컴팩트 프레임워크를 별도로 제공한다. 이러한 임베디드 시스템을 위한 개발 환경은 미들웨어용 개발환경을 지원하지 못하며, 유비쿼터스 컴퓨팅에 필수 불가결한 분산 환경을 지원하지 못한다. 이러한 문제점을 해결하기 위해 현재

ORB를 사용한 분산 미들웨어 프레임워크의 개발과 이 프레임워크를 이용한 통합 개발 환경의 연구가 수행되고 있다. 이러한 유비쿼터스 컴퓨팅 미들웨어 통합 개발 환경이 지원되면, 애플리케이션의 개발이 끝나지 않은 상태에서도 종합적인 시험이 가능할 뿐만 아니라 다양한 언어와 이기종 하드웨어 플랫폼의 지원으로 인한 소프트웨어 및 하드웨어 개발 일정 및 비용을 단축시킬 수 있는 장점을 가지고 있다

3. 그리드 미들웨어

그리드 컴퓨팅은 대용량 컴퓨터 자원을 효율적으로 분배하기 위한 서비스이다. 여기서는 네트워크로 연결된 거대 컴퓨팅 시스템뿐만 아니라 개인이 소유하고 있는 컴퓨터의 데이터, 컴퓨팅 파워, 애플리케이션, 스토리지, I/O 디바이스 등을 그리드를 이용하여 공간과 시간의 제약을 받지 않으면서 Grand Challenge Problem을 해결할 수 있도록 하는 기술이다. 이러한 그리드 기술이 발전할 수 있었던 것은 기존에 사용하고 있는 인프라를 새롭게 바꾸는 것이 아니라 현존하는 인프라를 좀더 효율적으로 이용할 수 있는 방법이어서 차세대 디지털 네트워킹 기술로 각광을 받고 있는 것이다.

세계적으로 그리드 미들웨어에 관한 연구는 Legion, Apples, NerSolve, JAMELIN, Ninf, Bricks, Globus 등이 있으나 현재는 소스 코드를 공개한 Globus가 가장 활발하고 광범위하게 연구되어지고 있다. 국내에서는 정보통신부와 KISTI가 협력하여 국가 그리드 구축사업을 하고, GFK와 KISTI가 그리드 미들웨어 기술 개발과 표준화에 대한 연구를 진행 중에 있다.

그리드 기술은 서로 다른 컴퓨터의 하드웨어 플랫폼과 운영체제에 대한 대비가 없어 멀티플랫폼을 지원하기에는 아직 미비한 실정이다. 또한, 그리드에 연결된 수많은 노드들로부터 발생하는 네트워크 트래픽 부하 처리를 위한 로드밸런싱 기술과 한 서비스 노드가 중지되었을 경우 다른 서비스 노드로 대치시킬 수 있도록 하는 동적 환경 유연성에 대한 연구가 필요하다.

4. OSGI(Open Services Gate Initiative)표준과 Home Networking 미들웨어

현재 홈 네트워킹 부분에서는 외국에서 개발중인 CEBus, HomePNA, Bluetooth, LONWORKS, HomeRF, UPnP, Jini, HAVi와 우리나라 기업인 삼성에서 연구 중인 VHN 등과 같은 다양한 표준들이 있다. 여기에서는 기술력이 가장 발전되어 있고 많이 사용되어지고 있는 UPnP, Jini, HAVi 등의 해외 기술과 삼성에서 연구 중에 있는 VHN에 대해 알아본다.

첫째로, UPnP(Universal Plug and Play)는 P2P를 지원하며 MS사에서 제공하는 분산모델을 적용한 미들웨어이다^[8]. 기존의 IP 네트워크와 HTTP 프로토콜을 사용하여 간단한 방법으로 홈 네트워크 기기의 제어를 구현하지는 목적을 갖고 하드웨어와 관련하여 Plug and Play와 자동 설정 기능을 제공한다. DHCP와 SSDP(Simple Service Discovery Protocol), GENA(General Event Notification Architecture), 그리고 SOAP(Simple Object Access Protocol) 등을 이용하여 HAVi처럼 Java Bytecode를 불러와 실행하지 않음으로써 차별성을 둔다. UPnP는 이미 검증된 웹 기술을 기반으로 홈 네트워크 기기간의 제어 모델을 구현하였다는 점이 가장 큰 장점이라 할 수 있다. 따라서, 하드웨어, 소프트웨어, 그리고 운영체제에 무관하게 동작이 가능하고 HTML을 이용하

여 간단하게 사용자 인터페이스를 제공하며, UPnP 포럼에 의해 기기와 서비스 타입이 잘 맞추어져 있다.

두 번째로 Jini는 홈 네트워크의 대표적인 미들웨어로 발전시킬 목적으로 Sun Microsystems사가 창안한 제어 모델이라 할 수 있다^[9]. Jini는 기존의 IP를 기반으로 하는 네트워크에 대한 자연스러운 확장성, Plug and Play 기능에 의한 간단한 시스템 구성과 실행 코드의 이동성에 의한 가변성, 그리고 Java 연관 제품 및 시스템과의 호환성 확보를 장점으로 보고 있다. 단점으로는 Jini 시스템에 매우 무겁고 수행 속도가 느리며 많은 양의 메모리를 차지함으로써 시스템의 단가가 높아지는 JVM을 도입했다는 것이다. 또한, Lookup Server에 너무 의존하여 홈 네트워크 시스템에 종종 발생할 수 있는 Lookup Server의 이탈이 발생 할 경우 전체 네트워크가 동작을 하지 않을 위험성도 있다.

세 번째로 HAVi(Home Audio Video interoperability)는 Sony사가 처음 제안한 홈 네트워크용 미들웨어로 IEEE1394 기술을 채택한 오디오 비디오 기기간의 실시간 데이터 전송물의 상호 호환성을 위한 미들웨어이다^[10]. 처음에는 Grundig, Hitachi, MEI, Philips, Sharp, Sony, Thomson, Toshiba 등을 포함하는 8개 회원사로 출발하였으나, 지금은 42개의 회원사를 두고 이 표준에 의해 오디오/비디오 제품을 개발

<표 2> 홈 네트워크 미들웨어

Feature	HAVi (Home Audio Video interoperability)	UPnP-MS (Universal Plug and Play)	Jini-SUN
Architecture	Peer-to-Peer	Peer-to-Peer	Client-Server
Programming Language	Java Binding	Independent	Java
OS & Platform	Independent	Dependent	Independent
Security	Java	IP Dependent	Java Dependent
Network	Non IP device	IP device	IP device
License	initial fee	Openq for members	Open license with fee for commercial use

하고 있다. HAVi는 AVC(Audio Visual Control) 커맨드를 사용하지만 미래에 등장할 기기도 자연스럽게 지원해주기 위해 DCM(Device Control Module)의 개념을 도입하고, IEEE 1394 기술을 적용하는 디지털 네트워크에 사용되는 Plug and Play를 지원한다. 즉, 각각의 기기는 RMI(Remote Method Invocation)를 이용하여 DCM 모듈로 기기 자신들을 표현하는 데이터를 전송하여 제어 커맨드가 제공하지 못하는 명령어를 이해하는 구조를 채택하였다. 또한, 다른 제조회사가 만든 기기들과도 통신할 수 있도록 투명성 있게 설계되었으며, 제어 신호 및 콘텐츠 등을 전송, 자바 바인딩을 통한 개방형 소프트웨어 API를 지원할 수 있다.

네 번째로 VHN(Versatile Home Networking)은 HAVi와 같이 IEEE1394 기술을 적용하면서 삼성전자에서 개발하는 TV 및 Set Top Box에 상호 제어용 소프트웨어로 개발한 IP를 근간으로 하는 홈 네트워크용 미들웨어이다. 삼성의 VHN은 여러 개의 이기종 네트워크의 인터넷을 지원하는 VESA Home Network Committee가 1999년 8월에 홈 네트워크의 미들웨어 솔루션을 포함하는 VESA Home Network Spec 1.0을 발표하면서 구체화되었다. VHN은 홈 네트워크에 접속되어 있는 모든 기기들이 공통된 인터넷워킹 프로토콜을 사용하여 통신을 가능하게 한다. 이 프로토콜들에는 네트워크 프로토콜을 사용한 통신용 프로토콜과 Network Layer 프로토콜은 물론 기기들을 네트워크 자율적으로 배치하는데 필요한 다른 프로토콜 등이 속한다. 그러나, 아쉽게도 VHN은 특정의 서비스를 전달하는 부분에 대한 표준이 아직 개발되지 않아 여기에 대한 연구가 필요한 실정이다.

OSGi(Open Services Gate Initiative) spec은 소규모 상업용 장치(small business appliances)를 이용하여 서비스 제공자(Service Provider)들이 가전기기에 접속해 인터넷 서비스를 제공할 수 있게 하는 오픈 인터페이스 규격 생성 및 유지보수를 목적으로 서비스 게이트웨이를 위한 API 집합을 정의하는 미들웨어이다^[11]. 지금

은 Java 기반의 API 규격을 개발하여 가정 가입자에게 다양한 서비스들을 제공하기 위한 방식을 개발하고 있다. 또한, 디바이스 관리 API, 시큐리티, 원격관리와 리소스를 포함한 end-to-end 방식의 서비스 구조를 정의한다. OSGi의 소프트웨어 스택은 Sun의 Java Virtual Machine 기술에 기반을 두고 있고, 규격은 응용 계층에 초점을 맞추고 있다.

OSGi는 1999년 3월에 Sun, IBM, Ericsson, Lucent, Motorola, Nortel 등 15개 회사를 중심으로 설립된 후 꾸준한 연구 개발을 통하여 많은 성과를 이루었으며, 현재는 우리 나라의 4 DHomeNet, ETRI, 삼성전자 등 약 80개가 넘는 기관이 참여 중이다. 그러나, OSGi 프레임워크는 분산 환경과 정보처리 상호운용을 제공하지 못하고, Residential에만 사용할 수 있는 클라이언트/서버 개념이므로 앞으로 이동성 및 분산 개념을 도입한 부분에 대한 연구가 필요하다.

5. J2M 플랫폼과 JVM, KVM

Sun사에서 개발한 모바일폰, PDA, 셋톱박스 임베디드 시스템용 자바 플랫폼은 J2ME(Java 2 Platform, Micro Edition) CLDC(Connected, Limited Device Configuration)/MIDP(Mobile Information Device Profile)와 CLDC/PDAP(Personal Digital Assistant Profile)가 있다. J2ME는 엔터프라이즈급의 J2EE와 데스크탑용의 J2SE, 그리고 스마트카드에서 사용되는 Java Card의 한 부분으로 구성되어 있다. CLDC는 nomadic 환경에서의 불안정한 네트워크 커넥션과 제한된 메모리, 그리고 낮은 프로세싱 파워를 가지고 있는 장치들을 위해 개발되었으며, 128KB부터 512KB까지의 메모리를 갖고 자바 플랫폼 및 자바 소프트웨어를 사용할 수 있도록 구성되어 있다. CLDC는 종류에 따라 두 가지로 구분되는데, 휴대 전화를 위한 CLDC/MIDP과, PDA를 위한 CLDC/PDAP로 나누어진다. 이에 더하여 J2ME 플랫폼이 등장하기 이전에 발표된 pJava(Personal Java)와 eJava(Embedded Java)는 임베디드

시장을 목표로 한 자바 플랫폼이다. 그러나, 주력 생산제품이 하드웨어 설계 및 제조가 아닌 Sun사는 하드웨어 업체들의 지지 없이 수 많은 임베디드 디바이스에 공통으로 적용할 플랫폼을 개발할 수 없다는 내부적인 한계와 기존의 J2SE 기반의 자바와 호환성을 유지할 수 없다는 한계점에 도달했다. 그래서 등장한 개념이 컨피규레이션과 프로파일로의 플랫폼 분할이다. 컨피규레이션이란 자바 가상머신과 코어 API들에 대한 명세를 의미하고, 프로파일은 표준 API집합에 대한 명세인 상위의 클래스 라이브러리이다. 컨피규레이션을 정의함으로써 메모리와 CPU 등의 성능과 같은 요구사항을 묶고, 각 디바이스들의 기능 혹은 시장의 요구사항에 맞추어 프로파일을 정의함으로써 플랫폼 통일성 및 다양성을 만족시킬 수 있도록 발전하였다.

경량 내장형 자바가상머신 분야에서는 Sun사가 1999년 JavaOne 개발자 컨퍼런스에서 J2ME의 컴포넌트로서 KVM(Kilobyte Virtual Machine)을 선보였다¹²⁾. <그림 3>과 같이 KVM은 CLDC 안에 포함되어 제공되는데 16/32-bit RISC/CISC 마이크로 프로세서와 컨트롤러를 위해서 제작 되었는데 셀룰러 폰, 호출기, PDA, POS 단말기 등에 적용된다. 그리고 자바 스택 위에서 총 160KB의 메모리 공간이 사용 가능하도록 구성되었다. KVM을 포함하고 있는 CLDC는 CLDC 1.0에서 CLDC 1.1이 개발됨에 따라서 기존에 지원하지 않았던 플로팅 포인트의 지원과 weak reference 및 이식성, 모듈성을 지원함으로써 한층 진보된 소형 디바이스 지향 미들웨어로 나아가고 있다. KVM에 비하여 고도의 최적화와 성능을 지원하도록 개발한 CLDC HotSpot Implementation은 효과적인 자원 관리 및 ARM 기반의 16/32-bit RISC 마이크로 프로세서와 컨트롤러, 192KB의 메모리를 지원한다.

6. 기타 미들웨어

기존의 미들웨어 시장은 개발자/서버 중심으로 구성이 되어 있었다. 플랫폼, 디바이스에 독립적

이지 않으며 서비스 제공 및 확장에 대해서는 고려하지 않고 개발되었다. 그러나, 차세대 미들웨어에서는 사용자와 서비스 중심으로 편의성 및 플랫폼에 독립적이고 PDA와 같은 디바이스에서부터 High-End 서버까지의 확장성에 중점을 두고 있으며, 미들웨어에서 제공하는 서비스를 응용프로그램, 웹 서비스와 같이 다양한 방향으로 확장 하는 것에 치중하고 있다. 즉, 사용자와 서비스 중심으로 변화되고 있으며, 이에 따라 ORB 개념을 기반으로 한 연구들이 이루어지고 있다. 이러한 차세대 미들웨어 관련 연구는 다음과 같다. 첫째, 프로토콜 프레임워크에 기반한 적응형 멀티미디어 ORB인 MULTE-ORB, 둘째, 컴포넌트 기반 reflective 미들웨어인 Open-ORB, 셋째, 분산 객체기반 미들웨어 플랫폼인 GOPI (Generic Object Platform Infrastructure), 넷째, 예측 가능한 실시간 QoS와 고성능을 특히 고려한 TAO 등이 있다. 이러한 미들웨어는 경량화, 최적화를 추구함과 동시에 마이크로 시스템의 제한된 자원을 사용하는 유비쿼터스 환경에 적용는데 무리가 있기 때문에 실시간성 및 경량화, 그리고 서비스 중심의 최적화에 관심을 두고 개발되어져야 한다.

V. 유비쿼터스 컴퓨팅 미들웨어의 문제점 및 향후 과제

일반적으로 지금까지의 미들웨어는 엔터프라이즈급의 응용 프로그램을 통합하는 것으로 고성능 시스템에서 동작하며, 클라이언트-서버 구조를 근간으로 하고 있다. 그러나, 우리가 지향하는 유비쿼터스 환경에서는 개발 언어 독립적이고, 객체 지향적인 미들웨어가 사용되는 CORBA와 DCOM 뿐만 아니라, CICS와 같이 기존의 메인프레임 환경에서 클라이언트의 접근을 허용하게 해서 고성능의 일관성 있는 처리를 허용하는 미들웨어, 자바환경에서 엔터프라이즈급의 애플리케이션을 용이하게 개발하고 전개할 수 있는 J2EE까지 모두가 다양한 디바이스들 간의 동적인 연결과 단

락을 반복하는 유, 무선 네트워크 환경에는 적합하지 않다. 위의 미들웨어를 유비쿼터스 환경에 적용하기에는 구체적으로 다음과 같은 문제점이 있다. 첫째로, 엔터프라이즈 네트워크상에서 실행되는 대부분의 애플리케이션들은 지속적인 연결(consistency)을 요구하며 컴퓨팅 환경이 자주 변하지 않아 동적인 상황인식(dynamic context awareness) 기능이 불필요하다. 둘째로는 자원관리 기능이 전혀 없는 것은 아니지만 유비쿼터스 컴퓨팅에서처럼 동적으로 변화하는 환경이 아닌 상당히 정적인 환경을 위한 것이다. 그리고 참여 클라이언트 또는 노드의 수가 제한되어 있어 자원관리가 일반적인 애플리케이션 수준에서 이루어진다. 따라서, 사용자를 둘러싸고 있는 주변 환경의 변화에 따라 변경된 자원을 관리해 주고 찾아주는 자원 관리 기능 및 서비스 디스커버리 기능을 찾아보기 힘들다. 셋째로는 엔터프라이즈급 애플리케이션은 획일화된 보안정책에 따라 보안방식이 결정되어 컴퓨팅 환경의 의미 또는 자원에 따라 보안체계를 변경하여 사용하는 유비쿼터스 컴퓨팅 환경에서 다양한 보안방식을 적용하기에 유연성이 상당히 결여되어 있다. 넷째로는 유비쿼터스 환경에서 사용될 프로세서 파워와 메모리 용량이 작은 마이크로 디바이스와 같은 하드웨어에 사용하기에는 포괄적인 서비스의 제공으로 인해 특성화되지 않고 너무 무겁(heavy)다는 단점을 가지고 있다. 마지막으로, 유비쿼터스 환경에서는 유무선 네트워크 장치간 뿐만 아니라 모바일 환경에 노출되어 있는 장치들 간의 호환성과 상호 운영성이 유지되어야만 한다.

이러한 문제는 유비쿼터스 컴퓨팅 환경이 수많은 센서와 디바이스 시스템들이 사용자의 생활에 스며들어야 하기 때문에 발생하는 것으로, 유비쿼터스 컴퓨팅 환경과 같은 고도의 분산 시스템을 위해서는 위의 다섯가지 문제점을 효율적으로 해결할 수 있는 미들웨어의 구현이 절실하다고 할 수 있다. 이러한 상황이 미래 유비쿼터스 지향 서비스를 개발하는데 매우 중요한 요건으로 작용하는데, 본질적인 유비쿼터스 컴퓨팅 서비스를

위해서는 각각의 디바이스 위치 및 상태 그리고 주변 환경과 같은 다양한 요소들의 조합에 의해 새로운 정보를 찾고, 공유된 정보를 사용하여 자원이 배분되기 때문에 향후 관련 기술의 유연한 진개를 위해서 아주 시급하게 개발해야 할 필요가 있다.

현재 유비쿼터스 미들웨어는 외부 액세스망(케이블, 광섬유, xDSL 위성 등), 유무선 홈네트워크(전화선, 전력선, USB, Ethernet, IEEE1394, WirelessLAN, HomeRF, Bluetooth 등)와 Ad-hoc 기반 실시간 에이전트와의 다양한 연동 기술을 기본으로 하는 홈 네트워크를 통합하고 응용하는 미들웨어 기술(OSGi)의 명세화가 시급한 과제이고, 이를 통한 기술 우위의 확보를 꾀해야 한다. 이에 더하여, 이기종 환경에 적합한 플랫폼 독립적이고 실시간 지원 및 컴포넌트 기반, 마이크로 하드웨어를 지원하는 마이크로 운영체제 구성요소와 커널, 그리고 경량 JVM 및 효과적인 유비쿼터스 자원의 관리와 서비스를 위한 자원/서비스 디스커버리의 기술 확보로 유비쿼터스 환경 지원 분산 미들웨어의 성능을 향상해야 한다. 이와 함께, 다양한 컴퓨팅 환경의 통합에 의한 QoS보장 멀티 객체간 협업 및 시스템/자원/환경 적응형 모델과 편의성 및 이동성 제공으로 인한 사용자의 요구를 충족시킬 수 있어야 한다. 더불어, 이동 환경에서 요구되는 보안 모델의 제시를 통한 이동환경 및 분산 환경에서 개인의 비밀 보장 기능이 추가되어야 한다. 그리고, 차세대 인터넷 기반 핵심 기술의 확보를 통하여 그리드 미들웨어 개발 기술에 이동성 및 편의성을 제공하기 위한 독자적인 모델을 확립하여 향후 차세대 인터넷을 선도할 이동 컴퓨팅 환경으로의 확장을 고려해 보아야 한다.

VI. 결 론

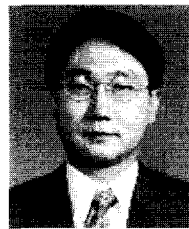
본 고에서는 유비쿼터스 환경에서의 미들웨어의 위치를 설명하고 현재에 개발되어지고 있는

미들웨어 프로젝트와 관련 기술에 대하여 살펴 보았다. 지금까지 개발되어진 기술을 보더라도 아직은 유비쿼터스 환경에서 이러한 미들웨어를 사용하기 위해서는 그 기능이 상당히 미비함을 볼 수 있었다. 이것을 해결하기 위해서는 다음과 같은 과제가 남아 있다. 첫째로, 신뢰성 및 네트워크에 독립성, 응용 간에 상호 운용성을 보장하고 사용자의 투명성을 지원하는 플랫폼 기술 개발을 위해 분산 운영체제 요소 기술, 분산 미들웨어 기술, 플랫폼 프로토타입 정립, 실시간 협업 시스템 기술에 대한 연구가 필요하다. 둘째로, 지능객체/엔티티간 실시간 데이터 처리 및 지능적/적응적/효율적 서비스 관리를 지원하기 위해 적응적 디스커버리 메커니즘을 지원하는 네이밍 기술^[3], Ad-hoc 환경에서의 실시간 통신 기술, 이기종간 플랫폼간의 인증 및 위임 기술을 필요로 한다. 셋째로는, 유비쿼터스 컴퓨팅/네트워크에서 주어진 상황을 인지하고 적응하여 사용자에게 편리하고 유용한 양질의 서비스를 제공하기 위한 지능형 환경/자원/사용자요구 적응기술, 환경 적응 Reconfiguration 기술, 지능형 예측/추론 기술 개발이 필요하다. 이러한 과제만 해결되어진다면 Mark Weiser가 말한 전자공간과 물리공간이 하나로 결합되고 모든 사물과 인간이 실시간으로 정보를 주고 받을 수 있는 유비쿼터스 환경 구축이 가능해지고, 인간 환경 개선의 새로운 패러다임이 공상 과학 영화나 상상에서 벗어나 실제 우리들 눈앞에 펼쳐지는 시기가 도래할 것이다.

참 고 문 헌

- [1] M. Weiser, "Some Computer Science Problems in Ubiquitous Computing", Communications of the ACM, July 1993, 75-84.
- [2] L. Kleinrock, "Nomadic computing: anytime, anywhere in a disconnected world", Mobile Networks and Applications, Vol 1(4), 1997.
- [3] Object Management Group (OMG), <http://www.omg.org>.
- [4] GAIA, <http://choices.cs.uiuc.edu/gaia/html/projects.htm>
- [5] Context Fabric, <http://guir.berkeley.edu/projects/cfabric/>
- [6] Oxygen Project, <http://oxygen.lcs.mit.edu/>
- [7] Eclipse, <http://www.eclipse.org>
- [8] The Univer Plug and Play Forum, <http://www.upnp.org/>
- [9] Jini Network Technology, <http://www.sun.com/software/jini/>
- [10] Home Audio Video Interoperability, <http://www.havi.org/>
- [11] OSGi Alliance, <http://www.osgi.org/>
- [12] KVM, <http://java.sun.com/products/cldc>
- [13] Naming and Addressing: URIs, <http://www.w3.org/Addressing/>

저 자 소 개



윤 희 용

1977년 2월 서울대학교 전기공학과(학사), 1979년 2월 서울대학교 대학원 전기공학과(석사), 1988년 8월 미국 Univ. of Massachusetts at Amherst, 컴퓨터공학과(박사), 1979년 3월~1984년 7월 : 금성정밀(주) 중앙연구소 선임연구원, 1988년 9월~1991년 5월 : Univ. of North Texas, 조교수, 1991년 6월~1999년 8월 : Univ. of Texas at Arlington, 부교수, 1999년 9월~2000년 8월 : 한국정보통신대학원 교수, 2000년 9월~현재 : 성균관대학교 교수, <주관심 분야: 모바일 컴퓨팅, 시스템 소프트웨어, 네트워크 스토리지>