

유비쿼터스 미들웨어 기술

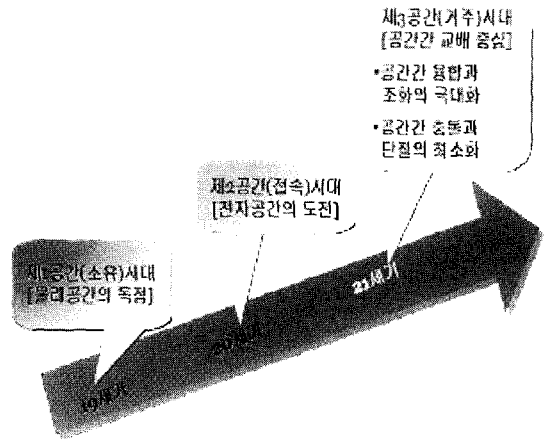
윤여봉* 윤희용** 정윤철*

◆ 목 차 ◆

- | | |
|-------------------------|--------------------------------|
| 1. 서론 | 5. 유비쿼터스 컴퓨팅 미들웨어의 문제점 및 발전 방향 |
| 2. 유비쿼터스 및 공존 패러다임 | 6. 맺음말 |
| 3. 유비쿼터스 환경에서의 미들웨어 위치 | |
| 4. 유비쿼터스 컴퓨팅 미들웨어 발전 동향 | |

1. 서론

과거로부터 현재에 이르기까지 인간들은 많은 변화를 보고 느끼면서 살아왔다. 그중에 대부분은 자연이나 환경 스스로가 변해온 것이 아니라 인간 스스로의 편의와 안락함을 추구 하기위해 인간이 변화시킨 것이 대부분이라 할 수 있다. 이러한 상황 속에 우리는 유비쿼터스 환경이라는 새로운 세계를 지금 맞이할 준비를 하고 있다. 지금까지의 IT 산업에 있어서 가장 큰 변화라고 여기어지는 유비쿼터스 환경은 인간의 조작에 의해 환경을 변화시키는 것이 아니라 환경이 인간의 상태와 주변 환경의 요소에 따라서 최적인 환경을 스스로 구축하는 것이다. 유비쿼터스 환경은 그림 1에서 보여주는 것과 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 제 3의 공간, 다시 말하면 물리적인 공간과 전자 공간을 하나로 융합화한 컴퓨팅 환경이다. 본고에서는 유비쿼터스 컴퓨팅에 대한 현 위치와 유사 컴퓨팅 패러다임에 대해 알아본 다음, 종류별 유비쿼터스 컴퓨팅 미들웨어 기술 및 현재 유비쿼터스 컴퓨팅이 가지고 있



(그림 1) 시대별 개발기술 패러다임 동향.

는 문제점들에 대하여 알아 본 후에 미들웨어 개발에 관한 향후 과제에 대하여 기술하고자 한다.

2. 유비쿼터스 및 공존 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

* 성균관대학교 컴퓨터공학과 대학원 석사과정

** 성균관대학교 정보통신공학부 교수

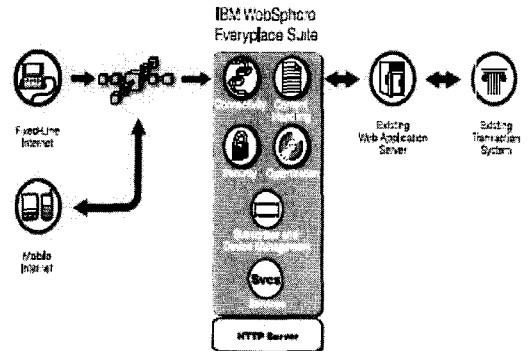
이용할 수 있는 인간, 사물, 공간 간의 최적의 컴퓨팅 및 네트워킹 환경을 구축함으로써 생활 속에서 자연스럽게 편리하게 컴퓨터를 사용할 수 있음을 의미하는 것으로, 1988 년에 미국의 마크 와이저에 의해 제창되었다[1].

유비쿼터스 컴퓨팅 환경을 구축하기 위해서는 사용자들이 흔히 접할 수 있는 사물 속이나 몸에 지니고 다니면서 언제, 어디서나 상호간에 정보를 주고 받을 수 있는 무선 통신망을 구축하여야 한다. 이러한 중요한 과제를 풀어 가기 위해서는 사용자에 따른 다양한 서비스를 제공하여야 하고 서비스들이 서로 간에 식별되어지고 에이전트에 의한 학습에 따라서 차별화 되어진 서비스를 제공하여야 한다. 또한, 불안정한 네트워크 환경에서도 스스로가 최적의 루트를 찾아서 안정적인 서비스를 제공받게 할 수 있는 고도 분산화된 네트워크 기술, 디바이스 간에 상태 정보를 수집하고 제어하는 서비스/자원 디스커버리 기술 등이 필요하다. 이와 함께 사용자와 유비쿼터스 에이전트를 보호하고 안심하고 서비스를 제공 받을 수 있는 에이전트 보호/인증에 관한 보안기술 등이 포함되어 안정된 유비쿼터스 환경을 제공할 수 있어야 한다.

이러한 유비쿼터스 환경이란 어느 순간에 갑자기 탄생한 것이 아니다. 많은 연구 인력과 산업 기술자들이 그동안에 개발된 다양한 컴퓨팅 기술을 최적의 환경으로 통합시킨 최종 단계인 것이다. 그동안에 유비쿼터스 환경을 만들기 위해서 증강현실(Augmented Reality), 착용 컴퓨팅(Wearable Computing), 노매딕 컴퓨팅(Nomadic Computing), 퍼베이시브 컴퓨팅(Pervasive Computing)과 같은 많은 컴퓨팅 패러다임들이 개발되었다.

증강 현실은 우리가 보고 있는 현실의 세계와 부가적인 정보를 가지고 있는 가상의 세계를 합쳐서 하나로 보여주는 것으로 가상 현실과 같이 현실 세계를 가상의 세계로 대치하는 것이 아니라 현실 세계를 보충해 주는 것이다. 그 대표적인 예로 GPS와 같은 사용자의 편의를 위한 가상의 세계를 도식화 해주는 교통 정보 시스템을 들 수 있다. 이는 Sentient Computing 이라 불리기도 한다. 착용 컴퓨팅은 현재 핸드폰과 PDA와 같이 소지한다는 개념보다는 의복과 같이 착용하고 다니는 환경을 말한다. 노매딕 컴퓨팅이란 [유

목의, 방향의]라는 뜻과 같이 장소나 디바이스들에 독립적으로 자신만의 정보 환경을 구축하여 일관된 방식의 정보를 제공 받을 수 있는 환경을 말한다[2]. 퍼베이시브 컴퓨팅은 유비쿼터스 컴퓨팅과 매우 유사한 개념으로 사용자가 인식하기 전에 미리 사용자의 요구 사항을 알아서 컴퓨터 스스로 사용자의 편의를 위한 최적의 환경을 구축하는 것을 말한다. IBM은 퍼베이시브 컴퓨팅을 현재 그림 2와 같은 Websphere Everyplace라는 솔루션을 통해 퍼베이시브 환경을 구축하는데 도움을 주고 있다[3]. 또한, IBM은 네트워크 환경으로 실현하고 있는 e비즈니스 환경을 자연스럽게 확장시킨 개념을 퍼베이시브 컴퓨팅이라고 정의한다.

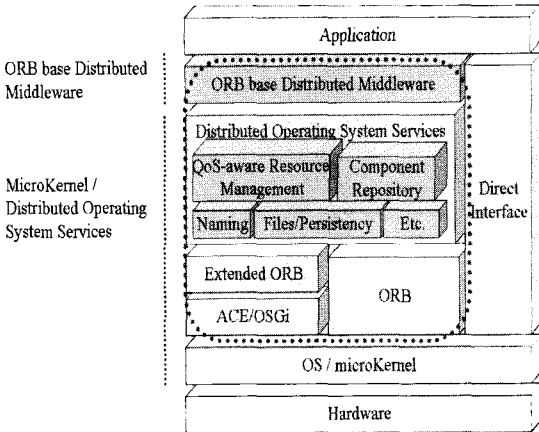


(그림 2) IBM Websphere Everyplace 구조.

3. 유비쿼터스 환경에서의 미들웨어의 위치

미들웨어란 그림 3과 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 있다[4].

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 기술 등이 필요하다. 또한, 이기종 환경에서 작동하는 다양한 센서, 디지털



(그림 3) 분산 운영체제와 미들웨어.

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 능동적으로 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 생활환경에서 사용되는 디바이스를 포함하기 때문이다. 이러한 디바이스들은 서로 간에 유기적인 통신을 하기 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 소프트웨어로 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 위해서는 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.

4. 유비쿼터스 컴퓨팅 미들웨어 발전 동향

현재 선진국에서 연구 중인 유비쿼터스 미들웨어 관련 프로젝트들을 보면 표 1과 같은데, 미들웨어의 중요성이 부각됨에 따라 많은 미들웨어들이 개발되었다. 이러한 각각의 미들웨어의 특색에 따라서 그 분류 또한 다양하게 나누어 볼 수 있다. 대표적으로는 PDA, 핸드폰과 같은 소형 디바이스뿐만 아니라 생활 환경에 산재되어있는 센서 등으로부터 이기종 하드웨어 및 플랫폼을 통합하여 데이터를 원활하게 연동할 수 있도록 지원 및 분산 처리하는 센서 미들웨어, 기존의 임베디드 소프트웨어 및 임베디드 OS가 가지고 있는 특징에 새로운 특정 기능 또는 서비스만을 위해 통합 개발 환경을 이루는 임베디드 소프트웨어 미들웨어, 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 그리드 컴퓨팅 미들웨어, CEBus, HomePNA, Bluetooth, LONWORKS, HomeRF, UPnP, Jini, HAVi, OSGi와 같

(표 1) 최신 유비쿼터스 컴퓨팅 미들웨어 관련 프로젝트(5.6.7).

프로젝트명	특징
CALAIS	<ul style="list-style-type: none"> • Context management에 관한 연구 • Context와 location awareness에 관한 연구 • 이 기종 장치간의 통일된 인터페이스 제공(event 기반) • Corba를 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 애플리케이션에 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

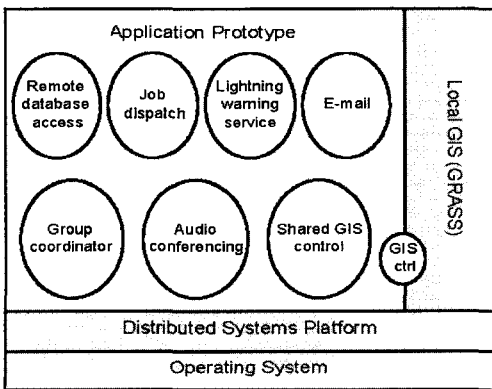
은 홈 네트워킹 기술의 표준을 정립하는 미들웨어, Sun사에서 개발한 모바일폰, PDA, 셋톱박스 임베디드 시스템용 자바 플랫폼지원을 위한 J2ME (Java 2 Platform, Micro Edition) CLDC (Connected, Limited Device Configuration) / MIDP(Mobile Information Device Profile) 와 CLDC/PDAP (Personal Digital Assistant Profile) 등이 있다.

이러하듯이 각각의 기능과 사용되어지는 분야에 따라서 미들웨어의 종류는 점차적으로 증가하고 있는 현실이다. 미국의 IEEE 분산시스템 온라인 커뮤니티에서는 이러한 미들웨어를 크게 네 가지로 나누고 있다[8]. 첫째는 개방성, 구성성, 재구성성을 수행하기 위한 미들웨어 시스템을 활용할 수 있도록 서술하는 리플렉티브 미들웨어 (Reflective Middleware)이다. 다시 말해 리플렉티브 미들웨어는 자신의 컴퓨팅 환경에서 실제적인 행동과 상태의 원인에 따라 스스로가 문제를 판단하고 적응시킬 수 있는 CCSR (Causally Connected Self Representation)을 통해 최적의 컴퓨팅

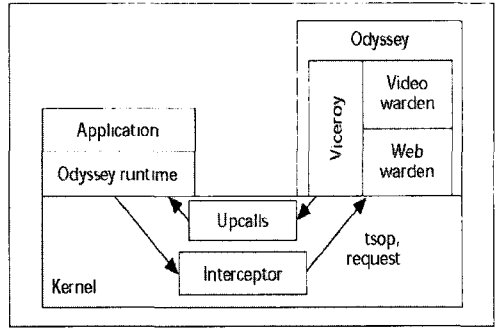
환경을 구축시켜주는 미들웨어이다. 둘째는 이벤트 기반 시스템을 만들기 위해 수행되는 개념, 설계, 수행능력, 서비스 또는 컴포넌트들의 응용에 중점을 둔 이벤트 기반 미들웨어 (Event-based Middleware)이다. 셋째로는 객체지향형 프로그래밍 패러다임에서 분산 시스템으로 확장된 객체지향형 미들웨어 (Object Oriented Middleware)이다. 마지막으로 OSI 네트워크 모델의 하위 레이어에서 효과적인 통신을 위한 패킷 패러다임에서 자연스럽게 파생된 메시지 지향 미들웨어(Message Oriented Middleware)이다. 이러한 미들웨어 중에서 주위 환경의 변화에 맞추어 스스로 변화하고 적응할 수 있는 자가 지식 성장이 가능하고 환경친화적인 유비쿼터스 환경에 가장 적합하고 그 발전 가능성이 있다고 판단되어지는 미들웨어는 리플렉티브 미들웨어로서 그 핵심 기능에 따라 어댑티브 미들웨어(Adaptive Middleware), 상황인지 미들웨어 (Context Aware Middleware), 리플렉티브 미들웨어로 나누어지고 있다.

4.1 어댑티브 미들웨어

어댑티브 미들웨어로는 MOST, Odyssey, K-Component 등이 있다. 첫 번째로 MOST는 Lancaster Univ 와 EA Technology가 공동으로 수행하고 있는 프로젝트로서 영국의 전력공사에 근무하고 있는 현장 근로자들에게 효율적인 업무를 위한 모바일 컴퓨팅 환경을 제공하고자 만들어진 프로젝트이다[9]. 이 프로젝트의 특징은 ANSAware 분산 시스템을 확장한 것으로 네트워크의 QoS의 변화에 따라서 관리되어지는 적응 기술에 초점을 두고 있으며 그림 4와 같은 구조로 구성되어 있다. 두 번째로 Odyssey는 미국의 미시건 대학과 카네기 멜론 대학에서 공동으로 수행하고 있는 프로젝트이다. 이 프로젝트의 특징은 모바일 정보를 가져오는 애플리케이션의 전송 범위를 지원해주는 프레임워크라고 말할 수 있는데, 시스템 안에서 모든 적응적 서비스를 결정하는 클라이언트 부분과 다양하고 효율적인 서비스를 판별하여 제공해주는 서버로 구성되는 미들웨어로서 그림 5와 같은 구조로 구성되어 있다[10]. 마지막으로 K-Component는 아일랜드 더블린의 Trinity College 분산 시스템 그룹에서 연구하는 미들웨어로서 보통의 동적인 소프트웨어 구조에서는 ADL(Architecture Description Language)와 AML(Architecture Modification Language) 그리고 Co-ordination Language를 혼합하여 사용하나 K-Component에서는 ADL을 대신하여 C++를 사용하였다. 이 미들웨어는 시스템 스스로 동적인 재구성을 지원하기 위하



(그림 4) MOST 시스템 구조.



(그림 5) Odyssey 구조도.

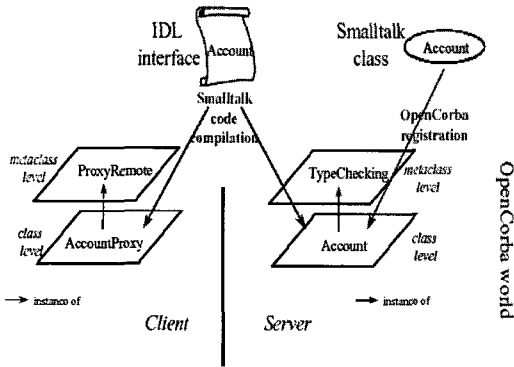
여 메타모델 구조(Architecture Meta-Model)와 주위 환경의 변화에 따라서 적응적으로 데이터의 크기를 줄이는 개체들로 구성되어 있다[11][12]. 지금까지 알아본 어댑티브 미들웨어에 대한 내용을 종합해보면 표 2와 같다.

(표 2) 어댑티브 미들웨어 분석 표.

분류	MOST	Odyssey	K-Component
기간	1993~1995	1994	2001
수행기관	Lancaster Univ & EA Technology	Univ of Michigan & CMU	Trinity College in Dublin
기반 요소	ANSAware	NetBSD	Architecture Meta Model
상용성	-	Y	-
핵심기술	QEX	Fidelity	Architectural Reflection

4.2 리플렉티브 미들웨어

리플렉티브 미들웨어로는 OpenCorba, Dynamic TAO 등이 있다. OpenCorba는 소프트웨어의 상태 정보나 실행 정책 등을 동적인 환경에서 사용자에게 적응시킬 수 있도록 하는 리플렉티브 오픈 브로커(reflective open broker)로서 클래스의 재사용성을 증가시키기 위해 분할되어진 메타클래스(Metaclass)를 사용하고, 시스템이 실행 중에도 상태의 변화에 대한 적응을 위하여 동적으로 메타 클래스를 변화시켜주는 프로토콜을

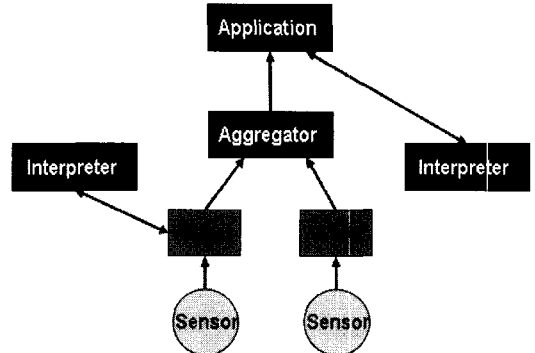


(그림 6) OpenCorba 통신원리.

제공한다. 이러한 OpenCorba는 클라이언트 쪽에서는 프록시 클래스, 서버 쪽에서는 템플릿 클래스를 생성 해주어 상호간 이기종 환경에서도 원활한 데이터 통신을 이루기 위한 구조로 되어 있다[13,14]. 그림 6은 이러한 프록시 클래스와 템플릿 클래스 사이의 통신에 대한 간략한 설명을 나타내고 있다. Dynamic TAO는 TAO ORB에서 확장된 미들웨어로서 각각의 모듈들은 C++를 사용하여 객체지향형으로 구현되었는데, TAO가 가지고 있는 쓰레딩 모델의 병행성, 역다중화, 스케줄링, 관리시스템 등을 모두 수용하고 있다. 또한, 외부 컴퍼넌트와 내부 컴퍼넌트 사이의 분별을 통하여 내부적으로 자체적인 조사와 재구성할 수 있는 시스템을 가지고 있다[15].

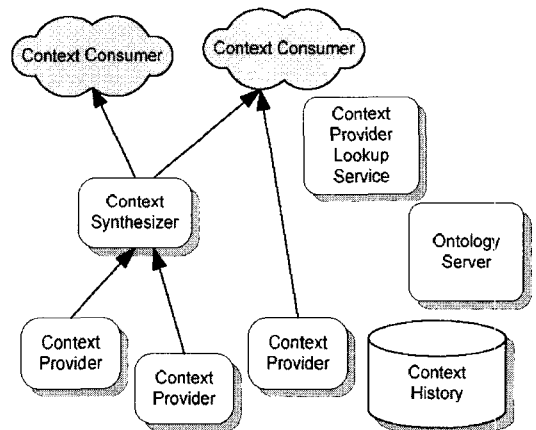
4.3 상황인지 미들웨어

상황인지 미들웨어로는 Context Tool Kit, Gaia와 RCSM 등이 있다. Context Tool Kit은 미국 캘리포니아 주립대 버클리에서 2000년도부터 시작한 Java기반 프로젝트로서 상황인지 애플리케이션을 지원하기 위한 일종의 툴킷이다. 이 미들웨어는 어떠한 상황에서도 분별력 있는 판단을 통하여 사용자가 원하는 정보 또는 관련 작업들을 제공해준다. Context Tool Kit은 하나의 상황에 대한 정보를 Widget이라는 객체에 은닉화를 시키고, 이러한 Widget과 애플리케이션화의 통신을 위한 게이트웨이와 같은 역할을 하는 Aggregator 객체로 구성되어있다. 또한, 하위의 상황 정보를 상위의 애플리케이션에서 원활하게 사용할 수 있도록 하



(그림 7) Context Tool Kit 객체별 상관 관계도.

기위한 Interpreter가 존재한다. 이러한 각각의 객체들의 관계를 보면 그림 7과 같다[16,17]. Gaia는 미국의 일리노이 주립대학교에서 2000년부터 시작한 프로젝트로서 미들웨어 프로젝트 중에서 가장 큰 규모를 자랑하고 있다. Gaia 프로젝트 중에서 Context Service 부분에 대하여 증점적으로 설명하면, 어떠한 상황의 전후 관계에 대한 정보를 취득 한 후에 그 상황에 대한 원인 분석을 하고 이에 적합한 의미론적인 상관관계를 보증하는 서비스로 다양한 원인 분석과 학습 메커니즘을 갖고 있는 에이전트를 사용한다. Context Service는 크게 그림 8과 같이 Context Consumer, Context Provider, Context Synthesizer, Ontology Server로 구성되어있다. Context Consumer 에이전트는 Ontology Server로부터 그들이 원하고자 하는 상황에 대한 구조를 가져온다. Context Consumer 에이전트는 Context Provider



(그림 8) Context Service 객체별 상관 관계도.

또는 Context Synthesizer로부터 정확한 질의 요청을 형성한 후에 그들이 원하는 상황을 가져온다. Context Synthesizer와 Context Provider 역시 그들이 알고 있는 모든 질의 요청에 대하여 Ontology Server로부터 요청하여 상황에 대한 정보를 가져온 후에 모든 정보를 다시 Context Consumer에게 제공하게 된다[18]. 마지막으로 RCSM(Reconfigurable Context Sensitive Middleware)는 퍼베이션 컴퓨팅을 위한 ad-hoc 네트워크 환경 요소와 상황인지 서비스를 제공하는 미들웨어이다. RCSM은 애플리케이션 안에 상황 인지적 요소를 접목 시켜 놓은 것으로, ORB를 통하여 상황 인지적 인터페이스와 상황 독립적인 이행요소들을 분리하여 차후에 사용자가 원하는 서비스의 요소를 보다 쉽게 서로 다른 임의의 상호 작용 요소들로 통합할 수 있게 하는 미들웨어인데 그림 9와 같다[19].

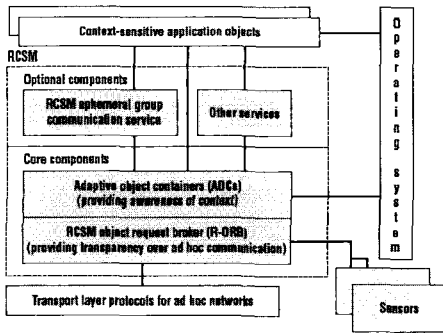


그림 9 RCSM 시스템 구조도.

5. 유비쿼터스 컴퓨팅 미들웨어의 문제점 및 발전 방향

일반적으로 지금까지의 미들웨어는 엔터프라이즈 급의 응용 프로그램을 통합하는 것으로 고성능 시스템에서 동작하며 클라이언트-서버 구조를 근간으로 하고 있다. 그러나 우리가 지향하는 유비쿼터스 환경에서는 개발 언어 독립적이고, 객체지향적인 미들웨어가 사용되는 CORBA와 DCOM뿐만 아니라 CICS와 같이 기존의 메인프레임 환경에서 클라이언트의 접근을 허용하여 고성능의 일관성 있는 처리를 할 수 있도록

하는 미들웨어, 자비환경에서 엔터프라이즈 급의 애플리케이션을 용이하게 개발하고 전개할 수 있는 J2EE 까지 모두가 다양한 디바이스들 간의 동적인 연결과 단락을 반복하는 유,무선 네트워크 환경에 적합하지 않다. 위에서 알아본 여러 미들웨어들을 유비쿼터스 환경에 적용하기에는 구체적으로 다음과 같은 문제점이 있다. 첫째로, 빠르게 변하고 있는 IT환경에 대다수의 일반 사람들은 어떠한 변화가 다가오고 있는지도 모르고 있다. 이러한 변화를 일반 사람들에게도 알릴 수 있는 다양한 홍보 활동, 전시회, 신문 그리고 방송활동을 통하여 일반인이 쉽게 다가 갈 수 있는 계기를 마련해야 할 것이다. 둘째로는 현재까지의 대부분의 유비쿼터스 미들웨어가 가지고 있는 문제점으로 지적이 되고 있는 보안에 관한 문제이다. 유비쿼터스 환경에서는 언제 어디에서나 어떠한 정보도 사용자가 쉽고 빠르게 접할 수 있기때문에 그만큼 사용자의 개인 정보는 어디에서나 노출되어져 있는 상태가 될 것이다. 우선적으로 이러한 보안관련 문제가 해결이 되어지지 않는다면 환경 친화적 인류 문명에 이바지 하는 유비쿼터스 기술 환경도 ‘빛 좋은 개살구’라는 오명을 벗기 힘들 것이다. 셋째로 현재의 미들웨어는 유비쿼터스 환경에서 사용될 프로세서와 작은 메모리 용량의 마이크로 디바이스와 같은 하드웨어에 사용하기에는 포괄적인 서비스의 제공으로 인해 특성화되지 않고 너무 무겁(heavy)다는 단점을 가지고 있다. 소형기기의 전력은 오로지 배터리에만 의존하고 있기 때문에 아무리 편리한 기술과 서비스를 제공한다 할지라도 시간상의 제약이 있다면 사용자들로부터 외면을 면하지 못 할 것이다. 이러한 문제를 해결하기 위해서는 산·학·연이 협동하여 가능한 한 모든 프로세스들을 저 전력화 시키는 기술 개발에 노력해야 할 것이다. 다섯 번째로 자원관리 기능이 전혀 없는 것은 아니지만 동적으로 변화하는 환경과 더불어 상당 부분에 있어서의 정적인 환경을 위한 것이어야 한다. 그리고 참여 클라이언트 또는 노드의 수가 제한되어 있어서 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. 마지막으로, 유비쿼터스 환경에서는

유무선 네트워크 장치들 뿐만 아니라 모바일 환경에 노출되어져 있는 장치들 간의 호환성과 상호 운영성이 유지되어야만 한다.

이러한 문제들은 유비쿼터스 컴퓨팅 환경하에 수많은 센서와 디바이스들이 사용자의 생활에 스며들어야 하기 때문에 발생하는 것으로, 유비쿼터스 컴퓨팅 환경과 같은 고도의 분산 시스템을 위해서는 위의 여섯 가지 문제점을 효율적으로 해결할 수 있는 미들웨어의 구현이 절실하다고 할 수 있다. 본질적인 유비쿼터스 컴퓨팅 서비스를 위해서는 각각의 디바이스 위치 및 상태 그리고 주변 환경과 같은 다양한 요소들의 조합에 의해 새로운 정보가 찾아지고, 공유된 정보를 사용하여 자원이 배분되기 때문에 향후 관련 기술의 유연한 전개를 위해서 아주 시급하게 개발해야 할 필요가 있다.

현재 유비쿼터스 미들웨어는 다양한 유·무선 통신 환경에 기반하는 실시간 에이전트와의 다양한 연동 기술과 홈 네트워크를 통합하고 응용하는 미들웨어 기술(OSGi)의 명세화가 시급한 과제이다. 이와 더불어 실시간 지원 및 컴포넌트 기반의 이기종 환경에 적합한 플랫폼 독립적인 마이크로 하드웨어를 지원하는 마이크로 운영체제 구성요소와 커널 기술 개발에 노력해야 할 것이다. 이러한 구성 요소들의 기술 개발을 위해 선행해야 할 작업으로 효과적인 유비쿼터스 자원의 관리와 서비스를 위해 자원/서비스 디스커버리의 기술 개발을 통한 분산 미들웨어의 성능 향상이 있다. 또한, 다양한 컴퓨팅 환경의 통합에 의한 QoS보장 멀티 객체간 협업 및 시스템/자원/환경 적응형 모델과 편의성 및 이동성 제공을 통해 사용자의 요구를 충족시킬 수 있어야 한다. 최종적으로는 지금까지의 컴퓨팅 패러다임들의 성공과 실패 사례분석을 통하여 필요한 기술은 받아들이고 필요 없는 부분은 과감히 배제하여 독자적인 모델의 정립을 통한 차세대 인터넷을 선도할 이동 컴퓨팅 환경으로의 확장을 고려해 보아야 할 것이다.

6. 맺음말

본 논문에서는 유비쿼터스 환경과 함께 미들웨어의

위치를 설명하고 현재에 개발 되어지고 있는 미들웨어 프로젝트에 대하여 살펴보았다. 다양한 미들웨어의 현재의 위치를 알아봄으로써 아직까지는 이러한 미들웨어들이 유비쿼터스 환경 구축을 위한 최적의 솔루션이라고 보기는 어렵다는 것을 알 수가 있었다. 이것을 해결하기 위해서는 다음과 같은 과제가 남아 있다. 첫째로는 이기종 운영체제간 상호 협력이 가능하며, 분산처리의 신뢰성, 네트워크의 독립성, 응용 프로그램 및 서비스간의 상호 운영성 및 투명성을 지원하기 위한 미들웨어기술 개발이 필요하다. 둘째로는 디스커버리 엔진, 응용 에이전트 기반의 인증 및 위임 기술 개발을 통한 지능 객체간 실시간 데이터 처리 관리 기술이 필요하다. 셋째로는 주어진 상황에 적응하도록 하기 위하여 변할 수 있는 상황을 시스템, 네트워크, 서비스로 분류하여 각각의 상황 변화에 맞추어 대응할 수 있는 Reconfiguration 기법을 개발해야 할 것이다. 이러한 과제만 해결되어진다면 전자공간과 물리공간이 하나로 결합되고 모든 사물과 인간이 실시간으로 정보를 주고받을 수 있는 유비쿼터스 환경이 우리들은 온 몸을 통해서 느낄 수 있는 시대가 올 것이다.

참고 문헌

- [1] M. Weiser, "Some Computer Science Problems in Ubiquitous Computing",
- [2] L. Kleinrock, "Nomadic computing: anytime, anywhere in a disconnected world"
- [3] IBM Pervasive Computing, <http://www-306.ibm.com/software/pervasive/index.shtml>
- [4] Object Management Group (OMG), <http://www.omg.org>
- [5] GAIA, <http://choices.cs.uiuc.edu/gaia/html/projects.htm>
- [6] Context Fabric, <http://guir.berkeley.edu/projects/cfabric/>
- [7] Oxygen Project, <http://oxygen.lcs.mit.edu/>
- [8] IEEE Distributed Systems Online, <http://dsonline.computer.org/>
- [9] A.Friday, N.Davies, "Developing Adaptive Application : The MOST Experience".
- [10] Brian Noble, "System Support for Mobile, Adaptive Applications".

- [11] Jim Dowling, Vinny Cahill, "Dynamic Software Evolution and The K-Component Model".
- [12] Jim Dowling, Vinny Cahill, "The K-Component Architecture Meta-Model for Self-Adaptive Software".
- [13] Nanbor Wang, Michael Kircher, "Applying Reflective Middleware, Techniques to Optimize a QoS-Enable CORBA Component Model Implementation".
- [14] Thomas, Ledoux "OpenCorba : a Reflective Open Broker".
- [15] Fabio Kon, Manuel Roman, "Monitoring, Security, and Dynamic Configuration with the dynamic TAO Reflective ORB".
- [16] Anind K. Dey, Gregory D. Abowd, "A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications".
- [17] Anind K.Dey Gregory D.Abowd, "The Context Toolkit : Aiding the Development of Context-Aware Applications".
- [18] Anand Ranganathan, Roy H.Campbell, "A Middleware for Context-Aware Agents in Ubiquitous Computing Environments".
- [19] Stephen S.Yau, Fariaz Karim, "Reconfigurable Context-Sensitive Middleware for Pervasive Computing".

● 저 자 소 개 ●



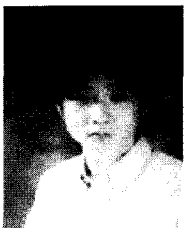
윤 여 봉

1995년 언남고등학교 졸업
 2001년 삼성 소프트웨어 멤버십 11기
 2003년 강남대학교 전자계산학과 졸업(학사)
 2004년~현재 성균관대학교 컴퓨터공학과 대학원 석사과정
 관심분야 : 모바일 컴퓨팅, 유비쿼터스 환경 미들웨어, 실시간 시스템



윤 희 용

1977년 서울대학교 전기공학과 졸업(학사)
 1979년 서울대학교 대학원 전기공학과 졸업(석사)
 1988년 미국 Univ. of Massachusetts at Amherst, 컴퓨터공학과 졸업(박사)
 2000년~현재 성균관대학교 정보통신공학부 교수
 관심분야 : 모바일 컴퓨팅, 시스템 소프트웨어, 정보보호 및 고장감내



정 윤 철

1996년 잠신고등학교 졸업
 2004년 건국대학교 컴퓨터과학과 졸업(학사)
 2004년~현재 성균관대학교 컴퓨터공학과 대학원 석사과정
 관심분야 : 모바일 컴퓨팅, 유비쿼터스 환경 미들웨어, LBS