

# U-City를 위한 유비쿼터스 컴퓨팅 미들웨어

정혜선\*

정창성\*\*

## ◆ 목 차 ◆

- |              |                     |
|--------------|---------------------|
| 1. 서론        | 4. 그리드기반 유비쿼터스 미들웨어 |
| 2. 그리드 컴퓨팅   | 5. 결론               |
| 3. 유비쿼터스 컴퓨팅 |                     |

## 1. 서론

U-City는 첨단 정보통신 인프라스트럭처와 유비쿼터스 정보서비스가 도구가 아닌 환경으로써 도시공간에 융합된 지능형 미래 도시로써 도시의 환경, 교통, 안전 분야에서 지리적으로 분산된 사이트간의 유기적인 협력을 효율적으로 지원해준다. 예를들어 사고가 발생했을 경우 어느 시간, 어느 지역에 있는지에 상관없이 사고 지역의 상황(Context)에 따라 고속 계산을 통한 실시간 데이터의 제공을 통하여 효율적으로 공간 및 자원에 대한 정보를 지능적으로 제공받을 수 있다. 이러한 유비쿼터스 환경 하에서 필연적으로 대두될 최대 화두는 “도처에 널리 퍼져 있는 수많은 컴퓨터로부터 데이터와 서비스 요구량이 폭발적으로 증가할 것이 자명하므로 이러한 방대한 양의 데이터와 서비스 요구를 어떻게 효율적으로 처리하여 어떠한 상황 하에서도 사용자에게 적합한 서비스를 제공할 것인가?”이다. 따라서 유비쿼터스 컴퓨팅 환경을 실현하기 위해서는 방대한 양의 정보를 효율적으로 처리하고 어떠한 상황에서도 사용자들에게 적합한 서비스를 자동으로 제공할 수 있는 표준화된 서비스 인프라스트럭처의 구축 관리 기술이 필수적이다.

이에 대한 해결책으로 지리적으로 분산된 고성능 컴퓨터, 대용량 데이터베이스 및 첨단 장비 등 다양한

컴퓨팅 자원을 초고속 네트워크로 연동함으로써 고속 연산, 대용량의 데이터 처리, 첨단 장비의 상호 공유 등을 가능하게 하고, 가상공간에서 협업 연구나 작업을 가능하게 해주는 그리드 컴퓨팅 기술을 사용자의 현재 위치, 시간, 사용자의 행동 및 작업 이력 등과 같은 사용자의 현재 상황 정보를 파악하고 분석하여 사용자가 현 상황에서 필요로 하는 서비스를 검색하여 구동시켜 주는 상황 인지(Context-aware) 컴퓨팅 기술에 연계시킴으로써 표준화된 서비스 인프라스트럭처를 제공할 수 있을 것이다. 따라서, 그리드 컴퓨팅 기술과 상황 인지(context-aware) 컴퓨팅 기술이 연계된 그리드기반의 유비쿼터스 미들웨어는 유비쿼터스 컴퓨팅의 기반 기술로써 유비쿼터스 컴퓨팅 환경을 구축할 수 있는 토대가 될 것이다.

본고에서는 우선 2장에서 그리드 기술의 발전 동향에 맞추어 표준화의 관점에서 기술하였고 3장에서는 유비쿼터스 컴퓨팅의 상황 인지 기술에 대해 해외 사례들을 살펴보고 4장에는 그리드기반 유비쿼터스 미들웨어의 아키텍처를 제시하고 5장에서 결론을 맺고자 한다.

## 2. 그리드 컴퓨팅

1998년, 미국에서 처음 그리드 개념이 제안된 이래로 글로벌 그리드 포럼(GGF: Global Grid Forum)을 중심으로 그리드 기술의 개발과 표준화 활동이 활발하

\* 고려대학교 전자공학과 박사과정

\*\* 고려대학교 전자공학과 정교수

게 전개되고 있으며, 국내에서도 2001년 5월, 정보통신부가 “차세대 인터넷 기반구조를 위한 국가 그리드 기본계획”을 수립하고, 2001년 10월, 그리드 포럼 코리아 (GFK: Grid Forum Korea)가 발족되어 해외 그리드 기술 및 표준화 동향을 조사, 분석해 국내의 각 연구 및 상용 기간망을 통합하는 등 본격적인 활동을 해 오고 있다. 이 활동의 일환으로 우리나라에서도 2005년 3월, 서울에서 GGF13이 열렸고, 한국의 그리드 기술을 세계적인 수준으로 한 단계 끌어올리는 동시에 국내그리드 관련 연구가 더욱 힘을 받는 계기가 되었다[1].

한편 그리드 기술의 표준은 현재까지는 글로버스 (Globus) 미들웨어를 중심으로 자리 잡아 왔지만, 이제는 웹 서비스 표준과 결합되어 통합된 기술 표준 형태로 발전하고 있다. 앞으로도 그리드 컴퓨팅 분야는 기술 표준의 통합과 컴퓨팅 모델의 퓨전 현상, 그리고 더 편리한 그리드 서비스를 제공하는 방향으로 발전할 것으로 전망된다. 그러한 대표적인 개방형 기술 표준의 예가 OGSA(Open Grid Services Architecture), WSRF(Web Services Resource Framework), SOA(Service Oriented Architecture) 등이다. OGSA는 인프라 자원의 공유를 위한 그리드 미들웨어 표준과 애플리케이션의 공유를 위한 웹 서비스 표준을 결합함으로써 그리드 기술을 보다 발전시키기 위한 개방형 기술 표준이다. WSRF는 2004년 1월, 글로버스 월드 2004에서 처음 발표되었는데, 그리드와 웹 서비스의 통합 아키텍처인 OGSA는 WSRF를 통해 차세대 그리드 표준 아키텍처로 한 걸음 더 발전하게 되었고 Globus Toolkit 4.0을 통해 완성되었다.

## 2.1 그리드와 OGSA

### (Open Grid Services Architecture)

2002년 2월, 캐나다 토론토에서 열린 GGF4에서 인터넷상의 애플리케이션 및 컴퓨팅 자원을 공유할 수 있는 통합된 개념의 OGSA가 처음 공개되었다. OGSA는 인프라 자원의 공유를 위한 그리드 기술과 애플리케이션의 공유를 위한 웹 서비스 표준을 상호 결합한 기술 개념으로 개방형 통합 기술 표준이다.

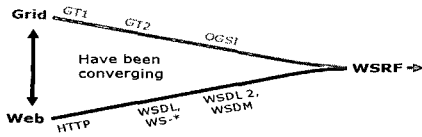
기존 글로버스 툴킷의 단점을 보완하기 위해서 글

로버스 미들웨어의 장점과 웹 서비스의 주요 요소인 XML, WSDL, SOAP, UDDI 등의 기술 표준을 통합하여 그리드 서비스의 개념을 확장하였다. 여기서 ‘서비스’는 네트워크에서 특정한 기능을 수행 하는 개체 (entity)를 의미한다. 굳이 객체(object)라는 용어를 사용하지 않고 서비스라는 용어를 사용한 것은 지금까지의 객체 관련 프로토콜이 시스템에 부하를 너무 많이 야기 시켰기 때문이다. 그런 점에서 웹 서비스는 기존의 분산 컴퓨팅을 위한 표준들인 DCE, CORBA, Java RMI 등과는 차별화되는 새로운 컴퓨팅 패러다임을 제공한다. 그 동안 업계에서는 널리 수용되어 온 웹 서비스 표준을 그리드 프로토콜과 결합시킴으로써, OGSA는 업계의 주도적인 표준으로 자리 잡아 왔다. 앞으로도 OGSA 기반의 그리드 표준이 보다 본격적으로 적용되어 간다면 개방형 표준에 기반 한 진정한 의미의 분산 협업 컴퓨팅이 구현될 수 있을 것이다. 한편, OGSA에 기반한 구체적 명세서(specification)가 OGS(Open Grid Service Infrastructure)인데, 이에 따라 2003년 7월, OGS 규약에 맞추어 구현된 최초의 글로버스 툴킷 3.0 버전이 발표되었다[2].

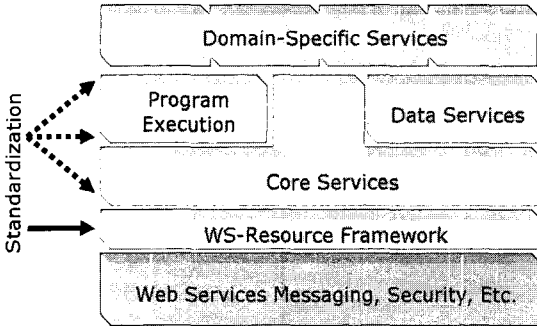
## 2.2 그리드와 WSRF

### (Web Services Resource Framework)

2004년 1월, 글로버스 월드 2004에서 WSRF가 발표되면서 그리드와 웹 서비스의 통합 아키텍처인 OGSA는 WSRF를 통해 차세대 그리드 표준 아키텍처로 다시 거듭나게 되었다. OGS는 W3C의 WSDL을 기반으로 만들어진 웹 서비스의 확장된 형태이기 때문에 그리드와 웹 서비스는 OGSA 발표 이후 급격히 융화되기 시작하였는데, 결국 OGS를 WSRF 형태로 재구성 (refactoring)하여 웹 서비스 스펙에 좀 더 가깝게 바꿈으로써 OGSA는 곧바로 웹 서비스 위에 자연스럽게 통합될 수 있게 되었다. 이렇게 하여 기업 환경에서 그리드 컴퓨팅의 적용이 좀 더 현실로 다가오게 되었다. OGSA는 앞으로도 WSRF와 결합되어 더욱 발전할 것이다. 개념에 있어서 WSRF가 좀 더 웹 서비스를 사용하기 용이하게 만들어졌으며 기존의 OGS를 크게 수정하지 않기 때문이다. WSRF는 현재 GGF의 OGS 위킹그룹에 의해 초안이 만들어진 상태이고



(그림 1) Convergence of Grid and Web Services



(그림 2) WSRF Architecture

OASIS(Organization for the Advancement of Structured Information Standards)를 통해 표준화가 추진 중이다. 이와 관련해 글로벌 얼라이언스는 WSRF에 기초한 글로벌 툴킷 4.0 버전을 2005년 5월 초에 발표하였고, 비로소 그리드는 실제적으로 웹 서비스를 기반으로 구축되고 동작하게 되었다. 따라서 용어상으로도 그리드 컴퓨팅은 그리드 서비스로 변경되어야 더 적합하게 되었다.

다음 (그림 1)은 그리드와 웹서비스의 컨버전스의 동향과 (그림 2)은 WSRF의 Architecture 이며 WS-Resource Framework 부분만 OGSA와 다르다[3].

### 3. 유비쿼터스 컴퓨팅

상황 인지(Context-aware) 컴퓨팅을 중심으로한 유비쿼터스 환경을 알아본다. 상황 인지 컴퓨팅 기술은 사용자의 현재 위치, 시간, 주변에 있는 다른 사람이나 정보가전 기기들, 사용자의 행동 및 작업 이력 등과 같은 사용자의 현재 상황 정보(contextual information)를 파악하고 분석하여 사용자가 현 상황에서 필요로 하는 서비스를 검색하여 구동시켜 주는 기술이다. 이러한 상황 정보는 센서 네트워크 내의 수많은 센서로

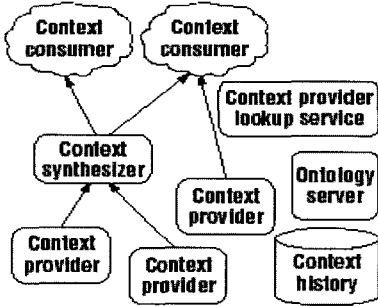
부터 수집된 자료들을 분석하여 파악할 수 있다. 현재의 컴퓨팅 환경 하에서는 사람들이 현 상황에서 자신에게 필요한 서비스가 무엇인지 우선 파악하고 그다음 이용 가능한 컴퓨터 혹은 기타 정보 기기들을 찾아서 직접 조작하여 필요한 서비스를 검색하고 구동시키고 있다. 상황 인지 컴퓨팅 기술의 큰 목적은 현재 사람들이 직접 하고 있는 위와 같은 일련의 모든 과정들을 사람들이 전혀 인식하지 못하게 대신 해주고자 하는 것이라고 볼 수 있다[4]. 상황 인지 컴퓨팅 기술은 유비쿼터스 컴퓨팅 환경에서 주요 서비스 기능들을 구현하고 있는 다양한 서비스들을 사용자의 간섭을 최소화하면서 효율적으로 사용할 수 있도록 도와주는 미들웨어이다. 이는 사용자 환경을 지속적으로 모니터링하고 이로부터 유익한 컨텍스트(context) 정보를 획득한 다음 획득된 컨텍스트 정보를 기반으로 사용자에게 효율적인 서비스들을 능동적으로 제공하는 것을 목적으로 하고 있다.

### 3.1 해외 사례

#### 3.1.1 Gaia

일리노이 대학 어바나 샴페인 캠퍼스에서 개발한 Gaia는 인간이 생활하는 물리적인 공간이 인간과 상호작용하고 인간은 그러한 물리적인 공간에 필요한 정보를 요청하고 이용 가능한 다양한 리소스를 통해 원하는 작업을 쉽게 수행할 수 있는 공간인 액티브 공간을 실현하는 것을 목표로 하고 있다. 액티브 공간이란 물리적인 공간과 소프트웨어 인프라가 함께 자연스럽게 융합되어 있는 공간을 말하며 이러한 공간은 이동 사용자를 지원하고 각 사용자는 다양한 리소스를 활용하고 다양한 장치를 사용하게 된다. 예로 한 사람이 회의실에서 회의에 필요한 발표를 하는 것을 가정할 때, Gaia는 회의실에서 이용 가능한 모든 출력 장치를 검색하고 이중 가장 적합한 대형 스크린을 선택하여 발표자료를 출력한다. 발표 중간에 참가자들의 PDA에 출력할 수도 있다. 또한, 슬라이드 출력 응용의 제어를 자신의 PDA로 할 수 있으며 회의실 내의 터치 스크린으로 할 수도 있다[5].

Gaia 커널은 컴포넌트 관리 코어와 5가지의 기본



(그림 3) GAIA의 Context Infrastructure

서비스(이벤트 관리, 컨텍스트 서비스, presence 서비스, space repository, 컨텍스트 파일 시스템)들을 제공한다. 이들 기본 서비스들과 응용은 컴포넌트 기반 분산 객체로 이루어져 있다. 컴포넌트 관리 코어는 이러한 Gaia 컴포넌트들과 애플리케이션을 동적으로 시스템에 로드, 언로드하거나 전송, 생성, 삭제하는 역할을 담당하며 CORBA 를 근간으로 하고 있다. 이벤트 관리자는 CORBA 이벤트 서비스에 기초하여 액티브 공간상에서 발생하는 이벤트들을 다른 컴포넌트에 분배하고, 컨텍스트 서비스는 응용이 특정 컨텍스트에 대한 질의를 수행하거나 특정 컨텍스트를 등록할 수 있도록 한다. 다음 (그림 3)은 GAIA의 Context Infrastructure의 각 컴포넌트들을 나타내고 있다.

새로운 자원이나 사람이 시스템에 추가되거나 삭제 되면 presence 서비스는 이를 인식하고 이벤트를 발생시킨다. 이러한 이벤트에는 인식된 자원이나 사람에 대한 정보를 포함하고 있다. Presence 서비스에 의해 발생된 이벤트는 space repository 서비스가 받게 되며 인식된 개체에 대한 설명을 담은 XML 정보를 저장 관리한다. 컨텍스트 파일 시스템은 개인적인 데이터 정보를 사용자의 존재 여부에 따라 자동으로 마운트/언마운트 하여 응용에 이용 가능하도록 해준다. 또한, 다양한 장치에 따라 동일 데이터의 적절한 타입 변환 서비스도 제공한다[6].

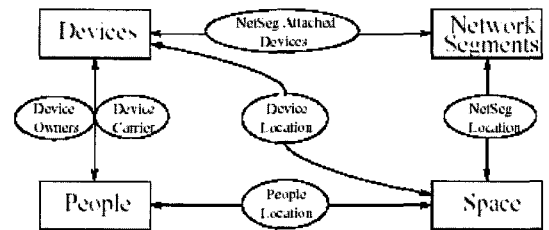
### 3.1.2 Aura

Aura의 주요 개념은 사용자-태스크-서비스-운영체제/네트워크-장치로 이루어진 레이어 구조에서 상위 레이어로부터의 요청을 시스템이 능동적으로 수용

할 수 있는 proactivity의 제공, 각 레이어에 요청된 요구사항의 적용, 리소스 사용량에 대한 자율적인 적응을 가능하게 하는 자가튜닝(self-tuning)이라고 할 수 있다. 예로, 공항에서 한 사람이 비행기를 기다리면서 작업을 수행하고 용량이 매우 큰 작업 결과를 메일로 회사 동료에게 전달하고자 할 때, Aura는 현재 사용자가 사용하는 무선 인터넷 환경을 이용하여 직접 메일을 보낼 것인지, 아니면 데이터를 먼저 압축한 후에 보낼 것인지, 좀 더 대역폭이 좋은 주변 환경을 이용하여 보낼 것인지 등을 결정하고 필요에 따라 사용자에게 부가적인 행동을 요청하고 작업을 수행하도록 한다[7].

Aura의 주요 구성요소는 Prism, Coda, Odyssey, Spectra이다. 위의 예에서처럼 사용자가 원하는 작업이 무엇인지를 인식하고 이를 시스템 전반에 표현해주는 Prism, 서로 다른 공간상에서 필요한 데이터를 공급해 주기 위한 분산 파일 시스템인 Coda, 현재 이용할 수 있는 자원의 모니터링과 그에 따른 적응을 수행하는 Odyssey, 이동 사용자 단말기의 빈약한 컴퓨팅 자원을 지원하기 위해 주변의 대형 컴퓨터의 인식 및 활용을 위한 원격 실행을 담당하는 Spectra로 구성되어 있다. 다음 (그림 4)은 Aura의 Context-aware application을 위한 Contextual service 모델이다.

Aura는 상황인지 뿐만 아니라 자원의 제약을 받는 이동 단말의 능력 배가를 위해 주변의 컴퓨팅 자원이나 데이터 서버를 활용할 수 있게 하는 Spectra나 현재의 자원에 따라 제공하는 서비스의 품질을 결정하게 하는 Odyssey를 통해 유비쿼터스 환경을 위한 유틸리티 컴퓨팅의 개념이나 적응형 컴퓨팅의 기본적인 기능도 수행하고 있다[8].



(그림 4) Aura의 context-aware application을 위한 contextual service model

### 3.1.3 RCSM

RCSM(Reconfigurable Context - Sensitive Middle - ware)은 애리조나 주립대학에서 연구 개발되고 있으며 컨텍스트와 서비스와의 능동적 결합, 즉석의 ad-hoc 통신을 요구하는 응용을 쉽게 설계하기 위한 미들웨어의 제공을 주요 목적으로 하고 있다. RCSM의 특징을 잘 보여주는 예는 다음과 같다. 강의실에서 강사가 강의를 시작하기 위해 프로젝션 스크린의 근처로 가고 강의실의 전등이 소멸되면, RCSM은 강사의 PDA에 있는 강의 자료를 학생들의 PDA로 전송한다. 강사의 강의를 끝난 후 학생들은 강의에서 발생한 특정 문제를 해결하기 위해 소그룹을 형성하고 토의를 진행한다. 강사가 특정 그룹의 근처에 오게 되면 RCSM은 강사의 PDA를 해당 그룹에 잠시동안 소속시키고 토의 자료들을 다운로드 받는다.

RCSM은 위의 예를 통해 알 수 있듯이 컨텍스트 적응형 응용과 ad-hoc 통신이 결합된 서비스를 제공한다. RCSM은 객체지향 개발 환경을 지향하고 있으며 CORBA와 매우 밀접한 관계를 갖고 있다. RCSM 응용들은 컨텍스트 적응형 객체들로 이루어지는데 이러한 객체들은 컨텍스트 적응형 인터페이스 부분과 서비스 구현 부분으로 구분된다. 컨텍스트 적응형 인터페이스는 컨텍스트와 이에 적용될 메소드의 조합으로 구성되며 컨텍스트 인지 인터페이스 정의 언어(Context-Aware Interface Definition Language : CA-IDL)로 기술된다. 이는 인터페이스 컴파일러에 의해 컴파일 되어 적응형 객체 컨테이너(Adaptive object Container: ADC)로 만들어진다. ADC는 획득된 컨텍스트 정보 중 관심 있는 컨텍스트를 발견하면 객체의 메소드를 실행한다. 또한, ORB(Object Request Broker)를 수정하여 컨텍스트 적응형 RCSM ORB(R-ORB)로 확장하여 응용 소프트웨어에 투명한 통신 환경을 제공하고 객체들을 대신하여 디바이스 및 서비스의 검색을 수행한다. RCSM은 컨텍스트 적응형 객체뿐만 아니라 기존의 CORBA, COM 객체들을 함께 혼용하여 사용할 수 있는 구조를 가지고 있다. 또한, RCSM은 성능상의 문제점을 해결하기 위해 ad-hoc 통신 등 성능에 민감한 부분을 임베디드 장치에 대하여 고성능을 달성하도록 FPGA(Field Programmable Gate Arrays)를 사용하여 하드웨어적으로 구현하였다[9].

### 3.1.4 Easy Living

1995년에 시작된 마이크로소프트(MicroSoft)사의 이지리빙(Easy Living) 프로젝트는 지능적 환경을 기반으로 하는 이동 컴퓨팅 기술 지원을 통하여 인간에게 가장 안락한 삶의 공간을 창조하겠다는 시험 프로젝트이다. 이지리빙 시스템의 사용자는 축소된 컴퓨터를 착용하고 이동하며 장치들을 원격 제어할 수 있어 물리적으로 장치에 접근할 필요가 없는 지능형 환경을 제공한다. 대부분의 컴퓨터는 사용자가 시스템을 사용하기 전에 로그인할 것을 요구한다. 이지리빙에서는 사용자가 실내에 들어오면 새로운 사용자에게 개인 ID를 부여하여 사용자는 컴퓨터에 로그인하지 않고도 웹을 사용하거나 전등을 켤 수 있다. 또한 로그인 패스워드나 실내의 지문인식기를 통해 사용자 자신을 인증하면 시스템은 인준된 사용자의 데스크탑 환경 제공과 개인 정보에 대한 접근을 허용한다. 뿐만 아니라 자동추적에 따라 다른 컴퓨터로 이동하더라도 로그인 절차 없이 동일한 환경으로 계속 작업할 수 있다. 사람들은 움직이면서 키를 입력하지 않고도 장치들과 상호작용할 수 있다. 예를 들어 걸어 다니는 관람자가 전시물을 관람하는 경우 위치감지에 의하여 관람자와 가장 가까운 디스플레이 장치에 전시물에 대한 설명이 자동으로 표시될 수 있다. 이렇게 마이크로소프트사는 사람의 움직임에 따른 입력을 전환할 수 있는 윈도우 기반 인터페이스를 연구하고 있다. 이지리빙 프로젝트는 이동 컴퓨팅과 지능적 환경이 함께 적용하여 유비쿼터스 컴퓨팅의 모든 기능이 실현될 것으로 추정하고 있다[10].

## 3.2. 온톨로지 기반 상황 인지 컴퓨팅 기술

유비쿼터스 환경은 “기계가 스스로가 상황 인지를 통해 정보의 의미를 이해하고 처리할 수 있는 거대한 정보의 공간”이라고 할 수 있다. 즉, 사람의 머릿 속에 있는 세상에 대한 이해를 컴퓨터 언어로 표현하고 이것을 컴퓨터가 사용할 수 있게 만드는 것인데, 사람과 기계 사이에 진정한 커뮤니케이션이 가능하기 위해서는 사람이 이해하는 수준으로 기계도 세상을 이해할 수 있어야 한다. 사람들이 세상을 이해하는 방



팅 레이어, 제공된 Context 정보에 따라 자동으로 서비스 실행을 가능하게 하는 유비쿼터스 메인 컴퓨팅 레이어, Computational 그리드를 사용하여 고속 실시간 처리 기능을 제공하는 응용 컴퓨팅 레이어, Access 그리드를 사용하여 협업 기능을 제공하는 협업 컴퓨팅 레이어로 구성되며 다계층 구조로 설계함으로써 모듈성, 재사용성, 확장성을 제공할 수 있을 뿐만 아니라, 추상화를 통해서 시스템의 복잡성을 감소시키고, 컴포넌트의 재사용성을 증가시키는 장점이 있다. 각 레이어에 필요한 모듈들을 간략히 살펴보면,

지능형 상황 인지(Intelligent Context - aware) 컴퓨팅 레이어는 특정 도메인 Ontology를 기반으로 Context를 정의하는 모듈과, 이를 기반으로 센서 데이터를 가공 처리하여 Context를 추출하는 모듈, 추론 엔진(inference - engine)을 통해 유용한 Knowledge를 생성, 관리할 수 있을 뿐 아니라 검색 및 정보 요구에 응답할 수 있는 프로세싱 모듈이 필요하다.

응용 컴퓨팅 레이어는 분산 자원의 공유를 통하여 방대한 데이터와 실시간 처리를 요구하는 응용문제에 대한 고속 계산을 제공하며, 이를 위해 데이터, 리소스, 실시간 정보를 관리하는 모듈이 필요하다.

협업 컴퓨팅 레이어는 효율적인 협업을 위해 Access 그리드상에서 데이터 및 응용프로그램의 공유를 지원하는 모듈과, Audio, Video 등의 Multimedia data를 사용할 수 있는 모듈이 필요하며 유비쿼터스하에서 효율적인 협업이 가능하도록 한다.

유비쿼터스 메인 컴퓨팅 레이어는 사용자들에게 투명한 지능형 서비스 인프라스트럭처를 제공할 수 있도록 Environment, Event, Task, Space 등의 핵심 서비스 모듈과, 이들이 서로 상호작용하면서 응용 컴퓨팅 레이어와 협업 컴퓨팅 레이어와 연동하여 하나의 단일화된 효율적인 유비쿼터스 서비스를 제공한다.

따라서, 그리드기반 유비쿼터스 미들웨어는 U-City의 환경, 교통, 안전 분야에서 고속 데이터 처리 기능과 협업 기능을 동시에 제공하면서 지능적인 인프라스트럭처를 제공할 수 있는 통합 유비쿼터스 컴퓨팅 미들웨어이므로 도시의 환경, 교통, 안전 분야에서 사고가 발생했을 경우 어느 시간, 어느 지역에 있는지 상관없이 사고 지역의 Context에 따라 고속 계산을 통한 실시간 데이터의 제공을 통하여 효율적으

로 공간 및 자원에 대한 정보를 지능적으로 제공받을 수 있도록 한다. 그리드기반 유비쿼터스 미들웨어는 재난 재해 방재시스템뿐만 아니라, 다양한 교통 상황에 대한 실시간 분석을 통하여 교통량에 따른 교통통제 및 교통상황 실시간 서비스 등의 지능형 서비스를 제공하는 교통 정보시스템과 수질, 대기오염 등의 데이터 분석 및 지능형 서비스를 위한 환경 부분 등에 적용 될 수 있다.

## 5. 결론

본 고에서는 지금까지 U-City 환경을 실현하기 위한 차세대 서비스 인프라스트럭처 구축 관리 기술로써 그리드 컴퓨팅과 상황 인지 컴퓨팅에 대하여 살펴보았다. 유비쿼터스 컴퓨팅 환경을 구축하기 위해서는 개방형 표준 기반으로 다양한 서비스를 효과적으로 실현할 수 있는 서비스 실행 환경이 필요하기 때문에 표준화된 공통 인프라스트럭처를 제공하는 그리드 컴퓨팅과 어떠한 상황에서도 사용자들에게 적합한 서비스를 자동으로 제공할 수 있는 상황인식 컴퓨팅 기술은 유비쿼터스 미들웨어의 핵심 기반 기술로 필수적이다.

본고에서 제시한 그리드기반 유비쿼터스 미들웨어는 유비쿼터스 환경 내의 수많은 컴포넌트들이 서로 상호 작용할 수 있는 공통 기반을 제공하며 즉, 유비쿼터스를 위한 장치와 서비스 들이 서로 연동되기 위한 공통 표준의 그리드 인프라스트럭처를 통해 하나의 단일화된 효율적인 유비쿼터스 서비스를 제공한다. 또한 다계층 구조로 설계함으로써 모듈성과 재사용성과 확장성을 제공할 수 있다. 유비쿼터스 컴퓨팅의 핵심 기반 기술로 그리드 컴퓨팅 기술과 상황 인지(context - aware) 컴퓨팅 기술이 연계된 그리드기반 유비쿼터스 미들웨어는 U-City의 유비쿼터스 컴퓨팅 환경을 구축할 수 있는 기술적 토대가 될 것이며, 또한 U-City의 환경, 교통, 안전 뿐만 아니라 다양한 산업계에서 발생하는 재해에 적용할 수 있고 기업 감시체계, 화재 감시체계, 의료 비상체계 등의 다양한 분야에 적용될 수 있기 때문에 산업적인 측면에서 매우 중요하고 파급효과가 크다고 할 수 있다.

## 참고문헌

- [1] 김양우, 한동헌 “미래 컴퓨팅 기술의 대통합, 그리드” 지식정보인프라 | 통권 23호 | 2006. 08\_ 23
- [2] The Open Grid Services Architecture  
<http://www.gridforum.org/documents/GWD-I-E/GFD-I.030.pdf>
- [3] The WS-Resource Framework  
<http://www.globus.org/wsrfl/specs/ws-wsrf.pdf>
- [4] 김창수, “Technologies for Construction and Management of the Service Infrastructure for Ubiquitous Computing” 전자통신동향분석 제19권 제5호 2004년 10월
- [5] <http://choices.cs.uiuc.edu/gaia/>
- [6] Manuel Romm, Christopher K. Hess, Renato Cerqueira, Anand Ranganathan, Roy H. Campbell, and Klara Nahrstedt, “Gaia: A Middleware Infrastructure to Enable Active Spaces,” IEEE Pervasive Computing, Oct.-Dec. 2002, pp.74-83.
- [7] <http://www-2.cs.cmu.edu/~aura/>
- [8] D. Garlan, D. Siewiorek, A. Smailagic, and P. Steenkiste, “Project Aura: Towards Distraction-Free Pervasive Computing,” IEEE Pervasive Computing, Vol.1, No.2, April-June 2002, pp.22-31.
- [9] <http://www.eas.asu.edu/~rcsm/>
- [10] S.A.N.Shafer, Easy Living,  
<http://www.research.microsoft.com/easyliving>
- [11] S.S. Yau, F. Karim, Y. Wang, B. Wang, and S.K.S. Gupta, “Reconfigurable context-Sensitive Middleware for Pervasive Computing,” IEEE Pervasive Computing, IEEE Computer Society Press, July-Sep. 2002, pp.33-40.
- [12] O. Storz, A. Friday, and N. Davies, “Towards ‘Ubiquitous’ Ubiquitous Computing: An Alliance with ‘The Grid,’” Proc. Ubisys: System Support for Ubiquitous Computing Workshop, in association with 5th Int’l Conf. Ubiquitous Computing(UbiComp 03), 2003.

## ○ 저 자 소 개 ○



### 정혜선

1992년 명지대학교 전자계산학과 졸업(학사)  
 2001년 고려대학교 전자컴퓨터공학과 졸업(석사)  
 2003년 ~ 현재 고려대학교 전자컴퓨터공학과 박사과정  
 관심분야 : Grid Computing, Ubiquitous Computing  
 E - mail : sepia5706@hotmail.com



### 정창성

1981년 서울대 전기공학과 졸업(학사)  
 1984년 Northwestern Univ. 전자계산학과 졸업(석사)  
 1987년 Northwestern Univ. 전자계산학과 졸업(박사)  
 1987년 ~ 1992년 포항공대 전자계산학과 조교수  
 1992년 ~ 1998년 고려대학교 전자공학과 부교수  
 1998년 ~ 현재 고려대학교 전자공학과 정교수  
 관심분야 : Distributed & Parallel Supercomputing, Grid Computing, Ubiquitous Computing,  
 Network Virtual Computing  
 E-mail : csjeong@korea.ac.kr