OSGi 기반 유비쿼터스 컴퓨팅 시스템 구조

OSGi Based Ubiquitous Computing System Architecture

정혜동, 함경선, 이형수 전자부품연구원
Hyedong Jung, Kyungsun Ham, Hyungsu Lee Korea Electronics Technology Institute

Abstract

Open Services Gateway Initiative(OSGi)는 WAN에서 로컬 네트워크와 장치에 다양한 서비스를 적용하는 것이 가능하도록 하는 기본 구조를 만들고자 많은 기업들이 참여하여 활동하는 컨소시엄이다. 본 고에서는 OSGi에서 제시하는 시스템 구조를 살펴보고 이를 유비쿼터스 컴퓨팅에 접목하기 위해 고려되는 사항과 다양한 환경에서 적용되기 위한 시스템 구조에 대하여 제안한다.

Keywords : Ubiquitous computing, Context aware, Open Service Gateway, Control middleware, OSGi, User interaction

I. Introduction

유비쿼터스 컴퓨팅은 이제 모든 컴퓨팅 환경을 언급할 때 빼 놓을 수 없을 만큼 밀접하게 고려해야 할 사항으로 인식되어지고 있다. 기존의 독립적인 플랫폼들도 상호 연동을 통해 보다 많은 역할을 할 수 있도록 하는 방안과 정해진 기능만을 수행하는 역할에서 지능적으로 판단하고 상황에 대처하는 지능형 플랫폼으로 융합되어 진화하고 있다.

이러한 시스템을 구성하기 위해서는 외부의 네트워크와 내부의 네트워크로 연결되는 지점에서의 서비스의 연계 및 융합을 담당하는 시스템 구조가 필요하며 적용되는 서비스가 끊임없이 이루어지도록 조정하여주고 플랫폼에 독립적으로 구성될 수 있는 유연함을 가져야 한다.

또한 로컬 네트워크와 디바이스들을 구성함에 있어 그 구성이 어느 한 곳에 종속적이지 않아 한 곳에 장애가 있어도 우회할 수 있는 다른 통로를 통해 서로간의 정보가 항상 전달되어야 하며 이러한 정보들의 수집은 연결되는 지점의 게이트웨이가 일반적으로 담당하게 된다. 이렇게 원격으로 적용되고 관리되는 서비스는 표준화가 이루어져야 많은 수의 회사가 참여하고 그 신뢰도를 높일 수 있다.[2]

2장에서는 이러한 시스템을 구성하기 위한 표준안인 OSGi에 대하여 살펴보고 3장에서 전

체 구성을 위한 구조를 그리고 4장에서는 구성된 시스템의 Use Case, 5장에서 전체적인 시스템 설계의 평가로 결론을 맺는다.

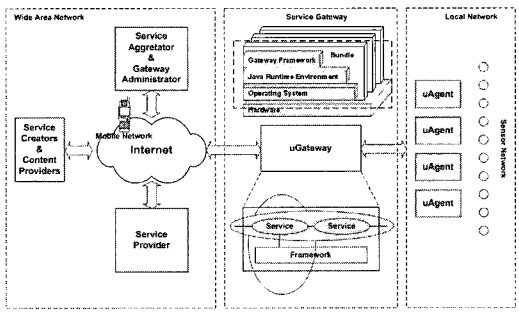
Ⅱ. The Open Services Gateway Initiative

Open Services Gateway Initiative(OSGi)[1]은 1999년 설립되어 사용자에게 다양한 서비스를 제공하기 위한 개방형 규격을 설립하는 일을 하고 있다. 이러한 규격 제정에는 통신 운영, 네트워크 제공, PC 업체 및 가전사, 자동차 회사등 다양한 분야의 산업계에서 활발히 참여하고 있고 최근의 유비쿼터스 컴퓨팅 연구와 맞물려 그 위상이 점차 높아지고 있다.

OSGi의 목적은 WAN에서 로컬 네트워크와 장치에 서비스를 적용시키는 것이다. 이러한 목적을 달성하기 위해 전체적인 시스템은 end-to-end의 해법을 이루는 구조를 제시하고자 Service Provider로부터 공급되는 서비스를 원활히 제공하는 구조를 연구하고 있고 이와 같은 전체적인 구조는 가정용 게이트웨이나 휴대전화 및 자동차와 같은 다양한 단말에 응용될 수 있다.

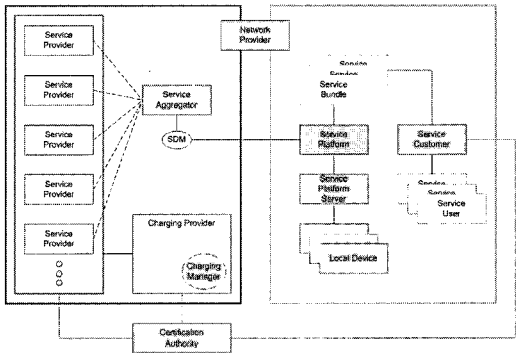
Ⅲ. Overall Architecture

전체 구조는 [그림 1]과 같이 Wide Area Network과 Local Area Network 사이에 Service Gateway가 위치하여 외부 네트워크와 내부 네트워크를 연계할 수 있는 교량 역할을 하게 되며 Wide Area Network는 서비스를 제작하고 컨텐츠 공급을 담당하는 Service Creators, Content Provider와 서비스 제공의 역할을 하는 Service Provider, 여러 Service Provider로부터 서비스를 받아 그 무결성을 입증하고 집합하는 Service Aggregator, 그리고 게이트웨이의 동작에 대한 책임을 지는 Gateway Administrator로 크게 구분지어 그 역할과 기능을 나누어 볼 수 있다. 또한 Local Network에는 네트워크의 끝에서 정보를 수집하고 서로간의 통신을 이루는 Sensor Network로 구성이 되고 이들이 수집한 데이터를 받아 적합한 형태로 가공하거나 게이트웨이로 전송하는 역할을 하는 uAgent가 그 네트워크를 이룬다. 그 중심에서 상호간의 서비스 연계를 담당하는 uGateway는 OSGi의 표준 스펙을 따라 여러 물리적인 기능을 담당하게되는 하드웨어 위에 운영 체제가 탑재되고 게이트웨이 프레임 워크를 플랫폼 독립적으로 운영하기 위해 자바 실행 환경이 설치된다.[1] 각각의 서비스는 번들의 형태로 제작되어 게이트웨이 프레임워크에서 동작하며 이러한 번들은 필요시에 탑재되거나 불필요할 경우 쉽게 제거 될 수 있으며 상호간의 데이터 교환을 통해 지능적 서비스를 이루어 낼 수 있는 구조가 된다.



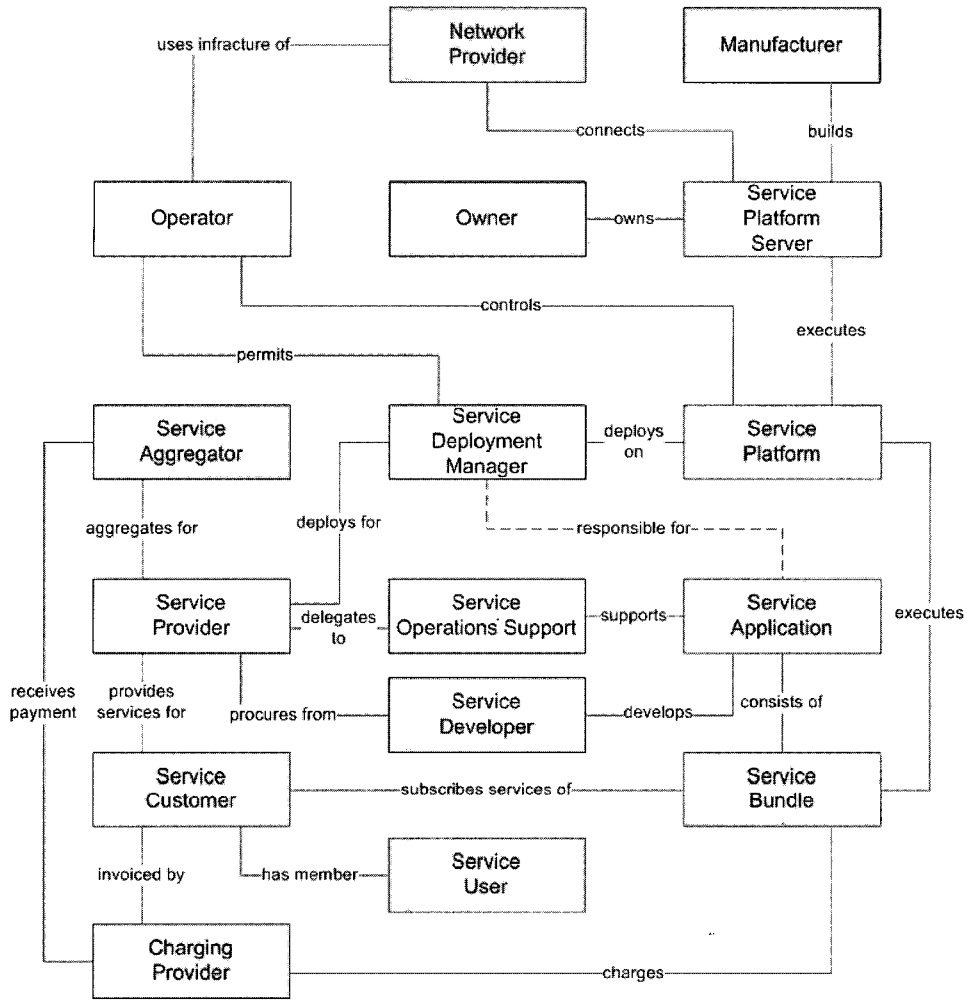
[그림 1] Overall Architecture

제안하는 구조는 OSGi 기반으로 설계되었으며 구조적인 설계와 함께 서비스를 제공하기 위한 전체 구성도는 [그림 2]와 같이 도식화 할 수 있으며 여기에서는 서비스가 Wide Area Network에서 Local Network와 Device로 흘러갈 때 서비스의 흐름과 이에 대한 과금 및 관리, 그리고 Local Network에서의 서비스 적용에 대한 내용을 나타낸다. 제작된 지능형 컨텐츠는 여러 Service Provider로부터 제작되어 Service Aggregator로 집약되며 이렇게 Service Aggregator에서 검증된 서비스는 Service Deploy Manager(SDM)을 통해 Local Network상의 Service Platform으로 전해진다. Service Platform Server는 서비스를 전해 받는 Service Platform을 포괄하는 하드웨어 및 시스템이며 Local Device에 받은 서비스를 적용하는 역할을 한다. Service Bundle은 서비스가 직접적으로 제공되는 형태이며 번들의 형태로 제공된 서비스는 Service Customer에 그 서비스를 원활히 제공하는 역할을 하게 된다. 서비스를 제공 받는 주체는 여럿이 될 수 있지만 그 과금의 주체는 주로 대표자로 이루어지게 되는데 예를 들자면 한 가정에 전화선이 공유되어 여러 명의 가족들이 전화를 사용한다고 해도 회선을 임대한 명의의 가족에게만 과금되는 것과 같이 제안하는 설계 구조에서도 서비스를 직접적으로 사용하는 Service User와 과금의 대상이 되는 Service Customer로 구분지어 질 수 있다. 제공 받은 서비스에 대한 과금은 Charging Provider에 의해 이루어지며 Charging Manager가 이를 판단하는데 Service Provider로부터 제공되는 서비스의 형태와 제공 받은 Service Customer 사이의 Certification Authority를 통해 종합적인 과금을 하게 된다. 이러한 구조는 실제 시스템이 적용되어 사용자에게 서비스를 공급하기 전 반드시 설계되어야 하는 부분이며 서비스 정책을 수립한 후 전반적인 시스템이 운영되어야 향후 발생할 수 있는 여러 서비스들의 융합에 대하여 대처할 수 있다.



[그림 2] 유비쿼터스 컴퓨팅 서비스 제공 구조

[그림 3]은 상기한 구조들이 집약되어 종합적인 서비스를 이루는 참고 구조이다. [4] OSGi 참고 구조의 가장 중요한 점은 잠재적으로 큰 네트워크의 서비스 플랫폼을 운영하는 모델을 기반으로 하는것이다. 따라서 서비스 플랫폼이 운영자에 의해 완전히 제어되고 여러 종류의 Service Provider들로부터 서비스들이 운영되는 것을 가정한다. 이러한 구조는 OSGi에서 강제적으로 적용시키는 표준안은 아니나 전반적인 구조는 서비스 정책을 수립하는데 아주 적합하며 이러한 모델을 기반으로 각각의 목적에 맞는 시스템을 구성했을 경우 타 OSGi기반의 서비스들과 다양하게 연계할 수 있고 어느 한 부분에 종속적이지 않으므로 유연하게 적용될 수 있다.



[그림 3] Reference Architecture

[그림 3]에서 보여지는 참고 구조는 기능별 블록으로 표현되어지며 각 블록간의 연관성은 블록을 연결하는 선에 기술되어 있다. OSGi 에서 제시한 위의 참고 구조는 상호간의 역할과 연계성을 명확히 보여주고 있으며 서비스의 흐름을 한눈에 살펴 볼 수 있도록 복합적인 구성을 간결하게 표현하고 있다. 서비스의 제공, 분배, 적용 및 사용에 대한 블록들로 이루어져 있으며 이를 구성하기 위한 플랫폼들의 위치를 정립한 구조이다. 위에서 기술했듯이 이러한 구조가 꼭 지켜져야 하는 것은 아니나 유비쿼터스 컴퓨팅 시스템을 제작할 때 선행되어야 하는 정책 수립을 하는 단계에 있어서 이러한 참고 구조는 큰 도움이 되며 각각의 블록에 대하여 정책을 수립하고 시스템을 설계한다면 향후의 시스템 검증에서도 상당히 유리할 것이다. 또한 검증된 시스템은 서비스 제공자나 사용자에게 그 특화성을 인증 받을 수 있으며 시스템의 기능 향상 시 양방향 모두의 만족을 이끌어 낼 수 있을 것이다. 또한 서비스는 어느 한

쪽에서 끝날 경우이거나 한쪽 방향으로만 흐르면 지능적인 시스템으로 설계되기가 어려우므로 [그림 3]과 같은 복합적이고 상호 방향성을 가지는 시스템은 지능적이고 적응적인 시스템을 구성하는데 적합하다고 할 수 있다. 시스템 구조의 기본적인 요소는 다음과 같다.

Business Driven: 운영주체의 관점(Business case)에서 참조 구조가 유도 되어야 한다.

Complete: 참조 구조는 vendor가 강건한 개발을 할 수 있도록 충분히 세부적이어야 한다.

Not Constraining: 서비스 플랫폼은 그 능력과 네트워크 환경이 다형적이므로 적용에 제한을 받으면 안 된다.

Open: 다수의 시나리오에 적용 가능하기 위하여 특정 시스템으로 설계되면 안 된다.

Not Normative: Normative specification으로 사용되어서는 안 된다.

또한 다음은 참조 구조를 구성하는 블록들의 설명이다.

Service Platform: Java VM의 인스턴스, OSGi 프레임워크, 운영되는 번들의 집합

Service Platform Server (SPS): 하나 이상의 Service Platform을 주관하는 하드웨어

Operator: 많은 수의 Service Platform에 대한 책임을 지는 기관

Service Application: Service User에게 유틸리티를 제공하기 위해 번들, 문서, 지원 소프트웨어의 모임 형태로 된 어플리케이션

Service User: Service Application으로부터 이득을 받는 사람

Service Provider: Service Application을 입수하거나 개발하고 이러한 어플리케이션을 Service Platform 위의 Service Deployment Manager를 통해 배치하는 기관

Service Deployment Manager (SDM): 하나 이상의 Service Provider에 대해 이를 배치하고, 부분적으로 Service Application들을 관리하는 시스템

Service Operations Support: Service Platform Server에 있지 않지만 Service Application을 실행하는데 필요한 지원 소프트웨어와 하드웨어

Service Aggregator: 서로 다른 Service Provider들의 서비스 어플리케이션의 무결성을 보장하는 것에 대해 책임을 지며 그것들을 하나의 형태로 통합하는 Service Provider

Service Developer: Service Application을 개발하는 기관

Manufacturer: Service Platform Server를 만드는 기관

Owner: Service Platform Server의 소유권을 가지는 사람이나 기관

Charging Provider: 계정 정보를 받고 Service Customer에게 통합된 청구를 하는 기관

Service Platform Identifier: Service Platform의 유일한 Identity

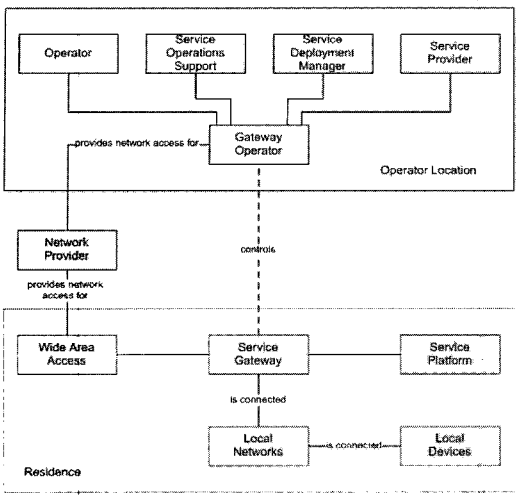
Service Customer: 과금에 이용되는 Entity

Network Provider: Service Platform들에게 네트워크 연결성을 제공하는 기관

Certification Authority: 인증 시스템, 개인, 그리고 기관들에 대한 증명을 관리하는 기관

다음에 도식하는 그림들은 몇 가지 서비스 게이트웨이 모델을 나타낸 것이다. 각각의 블록은 참조 구조에서 설명한 것과 같고 이 블록들이 실제 적용되는 모델별로 다양하게 구성이 됨을 보이는 다이어그램이다. 기본적인 모델과 산업계에 사용될 경우, 자가 운영 모델, 가상 게이트웨이 모델에 대한 구조를 보인다.

Basic Model



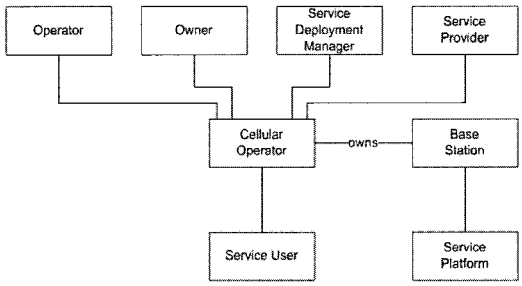
[그림 4] Basic Model

위 모델은 일반적인 형태의 게이트웨이를 기반으로 하는 시스템 구조이다. 그 역할에 따라 Operator 부분과 Residence 부분으로 나뉘며 Operator 영역에서는 서비스를 제작하고 제공하는 블록으로 구성되어 있으며 Network Provider를 거쳐서 Residence에 있는 게이트웨이에 서비스를 적용시키게 된다. Resicence로 내려오는 서비스는 Service Platform위에 적용되어 연결된 Local Network에 있는 Local Device에 그 서비스를 원활히 제공하는 역할을 담당하게 된다.

Industrial Model

- Assembly lines
- Firewalls in remote offices
- Point of sales terminals
- Energy Management systems for large buildings
- Alarm systems
- Fleet management

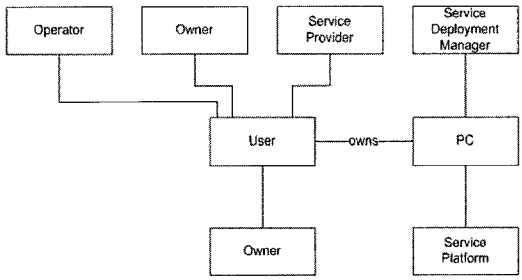
여기서는 산업용 구조 중에서 Cellular Network에서의 구현 예를 보인다. Basic Model에서의 Gateway역할을 Cellular Operator가 담당하여 서비스를 집합시키고 Cellular Operator가 가진 Base Station을 통하여 Service User에게 서비스를 제공하는 형태이다.



[그림 5] Industrial Model

Self-Managed Model

- Router / Firewall box
- WiFi base station
- Printer
- Private Branch Exchange (PBX)



[그림 6] Self-Managed Model

Self-Managed Model은 그 형태가 독립적으로 적용될 때 유용한 모델이다. Router, Firewall 혹은 WiFi base station등에 널리 사용되는 모델이며 PC와 같은 컴퓨팅 장치나 임베디드 시스템위에 구성되어 사용되며 특징적

인 것은 독립적 성격으로 인하여 자가 운영이 가능한 형태라는 것이다.

Virtual Gateway Model

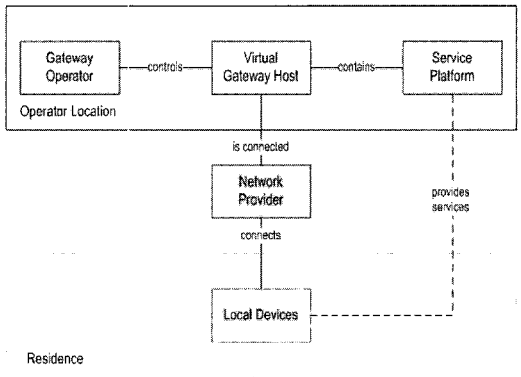
SPS가 외부에 존재하고 네트워크를 통해 서비스를 하는 형태의 구조이다. 이러한 형태는 Service Platform Server가 따로 설치되지 않아도 되는 특징을 가지므로 아파트와 같은 집단 거주 지역에서 거주자에게 장비 설치와 관리에 대한 불편을 끼치지 않으면서도 서비스를 제공 할 수 있는 장점이 있으며 보안에 특히 유리한 구조이다.

장점

- 운영자가 따로 없어도 된다.
- 거주자가 항상 연결되어있지 않을 경우 부담 경감
- 리소스 공유를 통한 효율성 확보
- 보안에 유리

단점

- 로컬 디바이스에 대한 연결이 어려워짐



[그림 7] Virtual Gateway Model

Ⅳ. Use Cases

본 고에서 기술한 전체 구조는 다양한 응용분야에 적용될 수 있으며 이러한 응용분야의 시나리오가 OSGi에서도 활발히 논의되고 있다. 다음에 기술하는 사용 예와 몇 가지 구조는 유비쿼터스 컴퓨팅 시스템을 구성하는데 있어서 전반적인 밑그림을 그리는데 도움이 될 것이다.

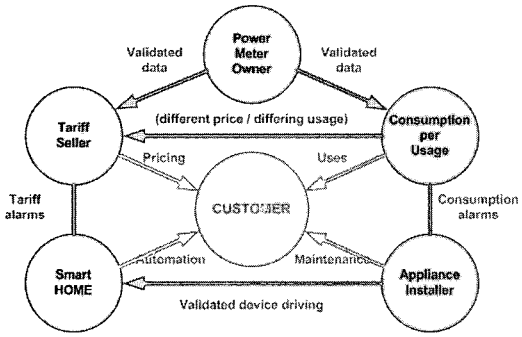
Personal Communications: 개인 통신의 복합적인 사용례로서 전화가 울리면 TV의 음량을 줄이는 등의 지능적 서비스가 가능하다.

Security: 게이트웨이의 중앙적 위치는 보안 구조를 적용시키는데 적합하다. 따라서 상황을 감시하고 이에 대한 경계를 Service Provider와 연계하여 복합적인 보안 체계를 구축할 수 있다.

Entertainment: Service Platform은 게임이나 음악, 비디오의 재생에 적합한 형태이므로 방송과 같은 컨텐츠들을 PDA와 같은 소형 기기들과 다양한 네트워크로 연결되는 기기들에 손쉽게 재분배 할 수 있는 중앙 통로가 될 수 있다.

Health Care: 독거노인의 경우 다른 어떤 사람들 보다 건강을 유지하기 위한 노력이 많이 든다. 유비쿼터스 컴퓨팅 시스템은 통신과 건강 경보 시스템의 연계로 이러한 문제를 손쉽게 해결 해 줄 수 있다. 예를 들어 냉장고의 문이 수일간 열리지 않거나 노인들의 몸에 착용된 센서 디바이스에서 검출된 맥박의 이상 등의 감지를 받은 게이트웨이는 병원과 연결되어 담당 의사의 호출을 할 수 있으며 응급 상황일 경우 응급 처치를 할 수 있는 구급대가 신속히 출동할 수 있다.

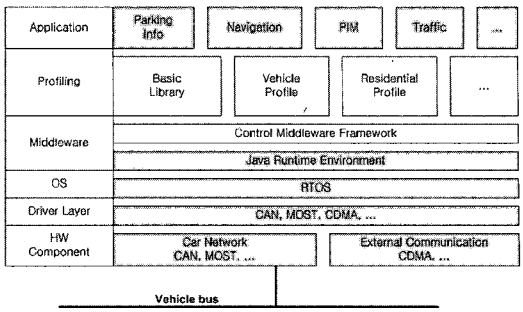
Energy Management: 에너지 관리는 가장 그 용도가 극명하게 나타날 수 있는 분야이며 개인의 에너지 절약으로 인한 비용 절감 보다 국가적인 에너지 절약에 따른 자원 관리가 가능해 지는 큰 효과를 볼 수 있는 분야이다.



[그림 8] Smart Metering

[그림 8]은 효율적인 에너지 관리를 위해 부하 관리를 통한 에너지 절약 정책을 수립할 수 있는 지능적 구조이다. 정책적으로 제시되거나 혹은 사용자가 설정한 부하를 넘을 경우 게이트웨이가 이를 판단해 불필요한 에너지 소비를 발생하는 장치를 끄거나 절약 모드로 전환할 수 있으므로 그 효과가 크며 이를 군집 형태의 가구나 기업으로 적용 시켰을때에는 국가적인 이익을 가져올 수 있다.

Vehicle: 유비쿼터스 컴퓨팅은 한 장소에만 국한되어 적용되어 지는 것이 아니기 때문에 이동 중에도 다양한 서비스를 받을 수 있도록 설계된다. 따라서 자동차 내부의 제어는 물론 외부 네트워크와의 연계로 집이나 회사의 장치들을 제어할 수 있으며 자동차의 이상을 제조사에서 실시간으로 파악하여 응급 서비스를 할 수 있다. [그림 9]는 OSGi 탑재 자동차 시스템 구조이다.[5]



[그림 9] Structure of Intelligent Vehicle

V. Conclusion

유비쿼터스 컴퓨팅 환경을 구성하기 위한 시스템적 요소들을 OSGi와 연계하여 살펴보았으며 이러한 구조들을 실질적으로 적용할 때 구성할 수 있는 시스템에 대한 모델을 제시하였다. 제안하는 구조는 서비스가 Wide Area Network에서 Local Network와 Device로 흘러갈 때 서비스의 흐름과 이에 대한 과금 및 관리, 그리고 Local Network에서의 서비스 적용을 쉽게 할 수 있으며 서비스를 위해 제작된 지능형 컨텐츠는 여러 Service Provider로부터 제작되어 Service Aggregator로 집약되며 이렇게 Service Aggregator에서 검증된 서비스는 Service Deploy Manager(SDM)을 통해 Local Network상의 Service Platform으로 전해지는 구조로 설계되었다. 또, Service Platform Server는 서비스를 전해 받는 Service Platform을 포괄하는 하드웨어 및 시스템이며 Local Device에 받은 서비스를 적용하는 역할을 하도록 설계되었다.

제시한 모델들은 어느 한 부분에 종속적이지 않도록 설계되었으며 이러한 모델을 응용하여 여러 서비스와 기기들을 연결하여 그 구조를 정립한다면 향후 상호 상승의 효과를 볼 수 있

을 것이다. 또한 실제 설계에 있어서 기본 모델 및 산업 보델, 자가 운영 모델, 가상 게이트웨이 모델등과 같은 여러 가지 설계 모델들을 응용하여 특화된 구조로 시스템이 설계된다면 전체적인 시스템 융합과 타 기기들과의 상호 연동성 및 다른 네트워크와의 연동 등이 수월해 질 수 있을 것이고 상호간의 수집 데이터를 쉽게 공유함으로써 유비쿼터스 컴퓨팅 환경을 보다 용이하게 이루어 낼 수 있을 것이다.

■REFERENCE

[1] The Open Services Gateway Initiative http://www.osgi.org

[2] Dave Marples, "The Open Services Gateway Initiative: An Introductory Overview", IEEE Communications Magazine, 2001, pp110-114

[3] Michael Condry, Ulrich Gall, Pierre Delisle, "Open Service Gateway Architecture Overview", Industrial Electronics Society, 1999. IECON '99 Proceedings. The 25th Annual Conference of the IEEE, Volume 2, 29 Nov.-3 Dec.1999

[4] OSGi Service Platform, Release 3, March 2003

[5] OSGi Alliance "OSGi Deployment and Traction", 2003 World Congress

Biography

정혜동(Hyedong Jung)

1998년 경희대학교 전자공학과(공학사)

2002년 경희대학교 대학원 전자공학과 (공학석사)

1998년 ~ 1999년 (주)카스 기술연구소

2002년 ~ 현재 전자부품연구원 유비쿼터스 컴퓨팅 연구센터, 전임연구원

관심분야 : 유비쿼터스 컴퓨팅 미들웨어, 멀티미디어 스트리밍 등

함경선(Kyungsun Ham)

1997년 광운대학교 컴퓨터공학과 (공학석사)

1999년 ~ 현재 전자부품연구원 유비쿼터스 컴퓨 팅 연구 센터, 선임연구원

관심분야 : 무선데이터통신, 무선 프로토콜, 내장형 시스템, 홈 네트워킹 등

이형수(Hyungsu Lee)

1989년 한양대학교 전자공학과(공학사)

2000년 아주대학교 컴퓨터공학과 (공학석사)

2002년 ~ 현재 성균관대학교 전전컴공학과(박사 과정)

1989년 ~ 1997년 LG전자 미디어통신연구소

1997년 ~ 현재 전자부품연구원 유비쿼터스 컴퓨팅 연구센터, 책임연구원

관심분야 : 유비쿼터스 컴퓨팅, 센서네트워크, 분산미들웨어 등