

Surrogate를 이용한 계층적 구조의 u-LifeCare 시스템 설계*

박상하, 조진성

경희대학교 컴퓨터공학과

okskpsh@khu.ac.kr, chojs@khu.ac.kr

Design of Hierarchical u-LifeCare System using Surrogate

SangHa Park, JinSung Cho

Dept. of Computer Engineering, Kyung Hee University

요 약

유비쿼터스 컴퓨팅 시대의 도래와 삶의 질 향상을 위한 개개인의 건강 유지와 증진에 대한 욕구는 유비쿼터스 라이프케어(u-LifeCare)산업의 발전을 촉진시키고 있다. 본 논문에서는 기존 헬스케어(HealthCare) 시스템에 관련된 연구들의 문제점을 분석하고 이를 해결하기 위한 새로운 u-LifeCare 서비스 시스템의 구조를 제안한다. 기존 헬스케어 시스템의 구조는 대부분 중앙집중형 Client/Server의 구조를 가진다. 이런 구조는 서비스 제공을 위한 모든 처리와 Data가 중앙 서버에 집중되므로 서버의 오버헤드 현상을 발생시킬 수 있으며 이는 사용자에 대한 서비스 응답시간(Response time)을 증대시키고 심할 경우 서버다운 현상을 초래할 수 있다. 제안하는 구조에서는 Surrogate를 동적으로 발견하고 Computing resource로 이용함으로써 이러한 문제점들을 해결하고 효율적인 서비스를 제공할 수 있다.

1. 서 론

네트워크 인프라의 발전과 임베디드 기술의 성장은 사용자가 언제 어디서나 인터넷에 연결된 컴퓨터를 사용할 수 있고 서비스를 제공 받을 수 있게 만들었다. 이런 유비쿼터스 환경이 도래함에 따라 가장 원초적인 인간의 욕구인 건강증진, 질병치료 서비스와 더불어 유비쿼터스 라이프케어(u-LifeCare) 산업이 각광받고 있다.

u-LifeCare란 기존의 HealthCare와 다른 개념의 서비스로 대상이 환자나 노인으로 국한된 것이 아니라 건강관리에 관심이 있는 모든 일반인을 대상으로 한다는 점에 있어 큰 차이점이 있으며, 기존의 질병의 치료와 관리 중점의 서비스와 달리 생활습관이나 패턴 등에서 오는 질병의 예방, 조기진단, 각종 의료시에 확대, 그리고 건강의 유지 및 증진에 그 목적이 있다.

먼저 기존의 연구들을 살펴보면 가정이나 병원 등 특정 도메인에 한정되어 플랫폼이 구축되어 있다. 이러한 구조는 언제, 어디서, 누구에게나 서비스가 가능해야 하는 u-LifeCare의 성격에 맞지 않다. 또한 대부분의 일반인의 생활범위는 직장이나 가정 외부에서 많은 시간을 보내기 때문에 효율적인 건강관리 서비스가 이루어 질 수 없다. 다음으로 플랫폼의 구조에 있어서 기존의 연구는 대부분 Client/Server구조로 되어있다. 이런 구조는 모든

Data의 처리와 서비스의 제공이 중앙의 Server에서 이루어지므로 소수의 사용자에 대한 간단한 서비스의 제공에는 문제가 없지만 다수의 사용자에 대한 복잡도 높은 서비스를 제공하기에는 무리가 있다. 즉, 서버에서 발생할 수 있는 오버헤드 현상은 결과적으로 사용자의 요청에 대한 응답시간을 증가시켜 서비스의 질을 떨어뜨리며 심하게는 서버다운 현상을 초래할 수 있다.

이런 문제점들을 해결하기 위하여 본 논문에서는 Surrogate를 이용한 계층적인 구조의 u-LifeCare 시스템을 제안한다. Surrogate란 Wired 또는 Wireless로 Internet에 연결되어 있으며, 사용자의 이동에 따른 Service discovery에 의해 동적으로 발견되어 사용되어질 수 있는 Computing resource를 말한다. 본 논문에서는 u-LifeCare 서비스를 위해 Surrogate를 이용하여 사용자에게 제공될 서비스를 수행한다. 이를 통해 서버에서 발생할 수 있는 오버헤드를 미연에 방지하고 Proactive 진단 시스템과 같은 복잡도가 큰 서비스의 수행을 Response time의 증가없이 수행하여 기존의 연구에서 발생할 수 있는 문제점들을 해결하고자 한다.

본 논문의 구성은 다음과 같다. 2장에서는 본 논문과 관련된 연구에 대하여 설명한다. 3장에서는 u-LifeCare 서비스 시나리오를 설명하고 4장에서 계층적 구조의 u-LifeCare 시스템의 전체 구조를 설명한다. 5장에서는 각 구성요소들의 기능을 정의하며, 마지막으로 6장에서

* 본 연구는 정보통신부 및 정보통신연구진흥원의 대학 IT연구센터 지원 사업의 연구결과로 수행되었음 (IITA-2006-(C1090-0602-0002))

결론 및 향후 연구방향을 제시한다.

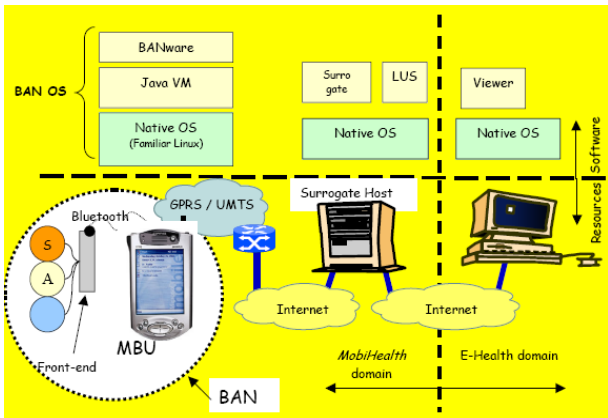
2. 관련연구

2.1 HealthCare System

유비쿼터스 헬스케어와 관련된 대표적인 프로젝트는 MobiHealth, 하버드 대학교의 CodeBlue 등을 들 수 있다.

MobiHealth[1]는 "Mobile Healthcare"의 약자로 EU에서 u-헬스케어 비즈니스 프로젝트로 수행되었으며 고위험도의 임산부, 만성 질환자, 심장 질환자 등을 대상으로택내 치료(Home care), 외상성장애 치료(Trauma care), 보행 모니터링(Ambulatory monitoring)에 중점을 두고 이를 통해 질병 판단 및 예측, 응급상황 대처 등의 서비스를 제공하는 플랫폼과 비즈니스 모델에 관한 연구를 진행하였다.

MobiHealth의 구성 요소 및 환경은 [그림 1]과 같다. BAN(Body Area Network)는 여러 센서들과 이들로부터 발생하는 Data를 수집하고 서비스를 제공하는 MBU(Mobile Base Unit)로 구성되어 있으며 Native OS, Java VM과 BANware로 구성된 BAN OS를 통해 동작한다. 사용자로부터 발생된 생체신호는 BAN을 통해 측정되고, 이러한 측정된 데이터는 원격지에 헬스케어 센터에 전달된다.

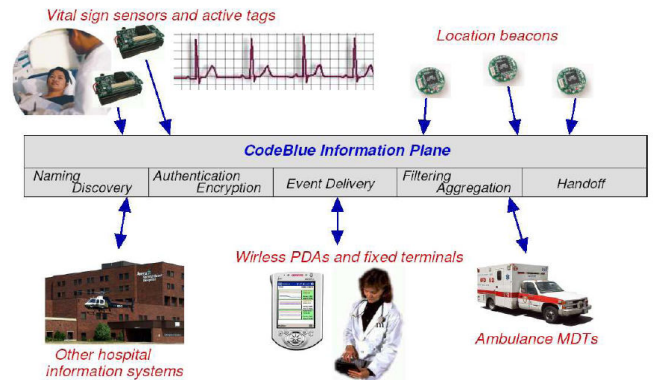


[그림 1] Components of the MobiHealth

또 다른 헬스케어 프로젝트로 미국 국가과학재단, 미국 국립보건원, 썬 마이크로시스템즈, 마이크로 소프트 등의 지원을 받아 하버드 대학에서 수행한 CodeBlue[2]가 있다. 이 연구의 목적은 Wireless Sensor Network 기술을 Medical Application에 적용시켜 병원 내외에서 환자에 대한 응급상황의 발견과 신속한 대처, 재난 상황에서의 긴급 의료활동, 발작환자의 재활치료 등에 활용하기 위해 추진되었다. 이를 위해 자체 개발한 무선 펄스 옥

시미터(Wireless Pulse Oximeter)와 쌍극(2-Lead) EKG를 이용한 센서노드를 통해 생체신호를 자동적으로 수집하여 기존의 병원 기록과 병합하여 실시간으로 환자의 상태를 모니터링하고 진단한다.

이러한 장비들은 심장 박동률(Heart Rate), 산소 포화도(SpO2), 심전도 데이터를 측정하고 단거리(100m) 내에 존재하는 PDA, PC, 노트북, 의료용 단말기 등에 무선으로 보내어 질 수 있다. 또한 이러한 센서들은 스스로 데이터를 처리할 수 있어 비정상적인 경우가 발생할 때는 알람을 울리거나 근접한 EMT(Emergency Medical Technician)나 진료보조자(Paramedic)에게 이러한 상태를 알릴 수 있도록 설계되었다.[그림 2].



[그림 2] CodeBlue Architecture

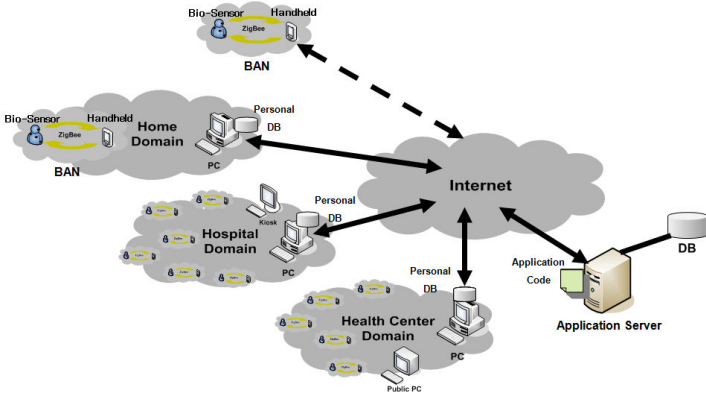
2.2 Surrogate Computing

반도체 산업의 발전과 PC의 대중화로 인해 데스크탑의 가격은 상당히 떨어졌으며 앞으로 더욱 낮아질 것으로 예측된다. 이런 현상으로 인해 현재에도 이미 많은 수의 컴퓨터가 카페나 공항, 버스 터미널 등의 장소에 공공의 사용을 목적으로 배치되어 있으며 그 수는 점점 더 증가하는 추세에 있다. Surrogate는 이렇게 공공의 장소에 Data staging server 또는 Compute server의 용도로 사용이 가능한 Computing resource를 의미한다. 또한 Surrogate는 대부분 Wired로 Internet에 연결되어 High-bandwidth의 네트워크 자원을 가지며 또 다른 큰 특징으로는 unmanaged, untrusted한 특징이 있다. 이와 같은 특징으로 인해 Surrogate는 유지보수 및 관리의 비용이 낮으며 이로인해 앞으로 Surrogate는 더욱 많이 확산될 것이다.

이렇게 도처에 존재하는 Surrogate의 자원들을 이용하기 위한 연구들이 위탁 컴퓨팅(Cyber foraging)[3], Public computing이라는 이름으로 연구되고 있으며 본 논문에서는 이들을 Surrogate computing으로 정의하고 이들을 이용하여 기존 HealthCare 시스템에서 발생할 수

있는 문제점들을 해결하고 Pervasive하고 Powerful한 서비스를 제공할 수 있는 u-LifeCare 시스템을 설계하고자 한다.

3. u-LifeCare 서비스 시나리오



[그림 3] 제안하는 u-LifeCare 서비스 구조도

3.1 Intra Body Area Network

사용자의 이동에 따라 항상 함께 이동하는 영역을 말한다. 내부적으로는 사용자의 몸에 부착된 Bio-Sensor로부터 주기적으로 또는 요구에 따라 Body Data가 발생하며, Handheld device는 기본적으로 이들 data를 수집하고 전달하는 기능과 Surrogate에서 처리된 서비스의 처리결과를 바탕으로 사용자 인터페이스를 제공하는 기능을 갖는다. 또한 Handheld device는 건물이나 스마트 오브젝트에 부착된 Sensor들의 Data또한 사용자의 이동에 따라 동적으로 발견하고 수집할 수 있다.

3.2 Home Domain

BAN영역이 가정 내부로 진입하면 Handheld device는 Service discovery를 통해 가정 내의 Surrogate를 발견하고 사용자 정보와 사용자가 제공받을 수 있는 서비스 목록 등을 바탕으로 Negotiation과정을 거친다. Surrogate는 이 정보를 바탕으로 Application server로부터 사용자에게 제공할 서비스에 대한 Application code를 다운받는다. Handheld device는 사용자의 요청에 따라 Surrogate로의 Remote execution 요청을 통해 Application을 수행하고 처리된 결과에 따라 사용자 인터페이스를 구성하는 방식으로 서비스가 이루어진다.

3.3 Other Domain

Other Domain은 병원이나 학교, 회사 등 가정 내가 아닌 외부의 장소를 의미한다. 이런 경우, 해당 도메인 내에는 많은 수의 사용자가 존재할 수 있게 되며 요청되는

서비스 량에 따라서 Surrogate에 오버헤드가 발생할 수 있다. 따라서 도메인 내의 각 Surrogate들은 서로간의 CPU처리량, RAM사용량 등의 동적인 Resource를 바탕으로 가용 Resource의 상태를 모니터링하며 리스트의 형태로 저장하고 있으며 오버헤드의 발생이 예측되면 적절한 Surrogate로의 Workload offloading을 수행한다.

3.4 OutSide of Domain

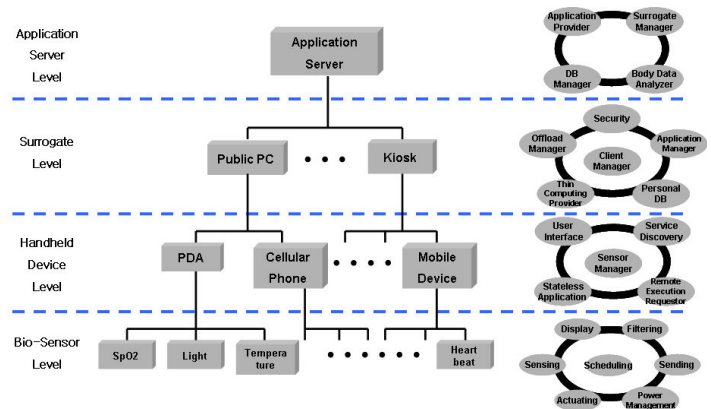
Service Discovery를 통해 Surrogate를 발견할 수 없는 상황을 말한다. Handheld Device는 Remote Execution의 수행 없이 자체적으로 수행할 수 있는 간단한 Application과 발생한 Body Data를 바탕으로 Emergency 상황에 대처하는 Application만을 수행한다. 하지만 Body Data의 처리는 실시간으로 Application Server의 통합 DB에 끊임없이 업데이트 된다.

3.5 Surrogate and Application Server

Application server는 사용자에게 제공될 모든 Application code와 통합 DB를 저장하고 관리하는 Back-end 서버를 의미한다. Surrogate는 자신이 관리하고 있는 사용자(Handheld device)에게 서비스를 제공해 주기 위하여 Application server로 Code와 personal DB를 요청할 수 있고, Application server는 이를 제공해 준다. Surrogate 범위 내에서의 사용자의 체류에 따라 Surrogate에는 해당 사용자의 Body data등의 정보가 저장되게 되고, 축적된 Data는 주기적으로 Application server로 업데이트 된다.

4. 계층적 구조의 u-LifeCare 시스템

제안하는 u-LifeCare 시스템은 4개의 계층적 구조로 이루어져 있으며 각각의 구성요소와 기능은 [그림 4]와 같다.



[그림 4] 계층적 구조의 u-LifeCare 시스템

5. u-LifeCare 시스템의 구성요소

제안하는 u-LifeCare 시스템의 각 컴포넌트에 대한 정의와 기능들을 설명한다. 제안하는 u-LifeCare 시스템은 Bio-Sensor, Handheld device, Surrogate, Application server로 구성되어 있으며 각 컴포넌트는 다음과 같은 기능을 갖는다.

5.1 Bio-Sensor

- ① **Scheduling:** Bio-sensor 내에 존재하는 태스크들을 FIFO또는 Priority 기반으로 Scheduling.
- ② **Sensing:** 주기적으로 또는 요구에 따라 값을 읽음.
- ③ **Filtering:** 예상 밖의 오차를 가진 Data 또는 에러를 포함한 Data들을 자체적으로 Filtering
- ④ **Sending:** Sensing된 값을 Handheld device로 전송
- ⑤ **Actuating:** Handheld device로부터 요청된 특정 Actuating 작업을 수행
- ⑥ **Power Management:** 처리할 작업에 따라 동적으로 각 Device들의 상태를 Idle로 변환하여 전체 배터리 사용량을 감소
- ⑦ **Display:** 필요에 따라 발생된 Body data나 Handheld device로부터 전송받은 Data를 Display

5.2 Handheld Device

- ① **Sensor Manager:** 사용자가 착용하고 있는 Sensor들에 대한 정보를 저장하고 관리하며, Body data의 수집과 Actuating 기능을 수행. Bio-Sensor 뿐만 아니라 고정적으로 설치된 Physical, Environmental sensor 또한 동적으로 발견하고 Data에 대한 처리를 수행.
- ② **Service Discovery:** 사용자에게 제공될 Application은 Handheld device에서 직접 수행되지 않고 사용자 주변에 위치한 Surrogate에서 수행. 이때 Surrogate를 동적으로 발견하기 위해 Dynamic surrogate discovery 기능을 Handheld device에서 담당한다.
- ③ **Remote Execution Requestor:** 사용자의 서비스 요청에 따라 그에 맞는 Application의 수행을 주변 Surrogate에 요청하고 결과를 받아서 처리.
- ④ **User Interface:** Remote execution의 결과를 바탕으로 사용자가 제공받는 서비스에 대한 UI를 처리.
- ⑤ **Stateless Application:** Remote execution 또는 통합 DB로의 접근이 필요하지 않은 간단한 서비스의 제공을 담당. Body data에 따라 Emergency 상황이라고 판단될 경우 즉각적으로 대처하는 Application을 포함.

5.3 Surrogate

- ① **Client Manager:** 현재 Surrogate가 Application의 처리를 담당하고 있는 Handheld device들의 정보와 제공 가능한 서비스를 관리.
- ② **Offload Manager:** 주변 Surrogate들의 사용 가능한 Resource량을 실시간으로 모니터링하며 리스트형태의 정보로 유지함. 이때 사용자의 수가 많아지거나 요청이 증가하여 오버헤드가 발생하는 것을 방지하기 위해 리스트 내의 다른 Surrogate로 Offload를 수행.
- ③ **Application Manager:** 사용자가 요청한 서비스나 제공 가능한 서비스의 수행을 위해 Application server로부터 Application code를 요청하고 관리.
- ④ **Remote Execution Provider:** Handheld device의 요청에 따라 Application code를 수행하고 결과를 Remote execution requestor로 처리결과를 전달.
- ⑤ **Personal DB:** 사용자의 출입에 따라 Application server의 통합 DB로부터 Surrogate로 필요한 이전의 DB를 가져오거나 체류기간 동안 Surrogate에 임시 저장된 DB를 업데이트하고 삭제.

5.4 Application Server

- ① **Surrogate Manager:** 사용자에게 서비스를 제공해 주고있는 Surrogate들의 리스트를 저장하고 관리.
- ② **Application Provider:** 제공 가능한 모든 서비스에 대한 Application code를 저장하고 관리하며, Surrogate의 요청에 따라 Application code를 전송.
- ③ **Body Data Analyzer:** 장기적으로 축적된 Body data를 바탕으로 특정 패턴의 검출이나 신체변화정도, 건강지수 등을 분석하여 해당 Application에 제공.
- ④ **DB Manager:** 전체 데이터베이스를 저장하고 관리. 각각의 Surrogate가 요구하는 사용자에게 대한 Personal DB의 요청과 업데이트에 대한 처리를 담당.

6. 결론

본 논문에서는 사용자의 이동에 따라 도처에 존재하는 Surrogate들을 동적으로 발견하고 이용하는 계층적 구조의 u-LifeCare 시스템 구조를 제안하였다.

제안한 구조는 중앙 집중적인 기존의 HealthCare 시스템 구조와 달리 유희자원인 Surrogate로의 분산처리를 통해 기존 시스템들의 문제점을 해결하고, 또한 앞으로 다양하게 제공될 u-LifeCare 서비스의 수행을 더 빠르고 효율적으로 수행할 수 있을 것이다.

7. 참고문헌

- [1] www.mobihealth.org/

[2]<http://www.eecs.harvard.edu/~mdw/proj/codeblue/>

[3]R.Balan, J.Flinn, M.Satyanarayanan, S. Sinnamohideen, H.Yang, "The Case for Cyber Foraging." In Proceedings of the 10th ACM SIGOPS European workshop, Saint-Emilion, France, September 2002.