USN 미들웨어 기반의 헬스 케어 모니터링 시스템에 대한 연구

김의창*, 강해성**

*동국대학교 정보경영학과, **동국대학교 전자상거래협동과정 기술전공

A Study on the Healthcare Monitoring System based USN Middleware

Kim Yei-Chang, Kang Hae-Sung

Dongguk University

E-mail : kimyc@dongguk.ac.kr, benjamin0705@naver.com

요 약

USN은 하드웨어와 유·무선 통신기술이 서로 네트워크로 연결되어 사물이나 환경의 정보를 획득하고 가공해 유용한 정보로 창출하는 것을 목표로 하고 있으며, 그 중 u-Healthcare는 의료 서비스 질 향상이라는 필요성에 의해 이슈가 되고 있다.

본 논문에서는 이 기종 센서 네트워크에서 획득한 데이터를 추상화하고 상황에 맞는 서비스를 제공하는 지능형 미들웨어를 설계하였다. 설계된 미들웨어를 사회적 이슈가 되고 있는 IT의료 분야에 적용해서 헬스 케어 모니터링 시스템을 구현했다. 센서를 통해 온도, 습도, 조도, 심박 수, 체온을 측정해서 실시간으로 관리하고 모니터링 하는 기능을 지원한다. 또한, 실시간으로 전달되는 바이오 정보를 통합 관리하는 부분에 초점을 맞추어 연구를 진행했다.

1. 서론

오늘날 유·무선 정보통신 기술의 발달은 다양한 분야에서 적용되고 있는데 가장 큰 특징은 무선기술의 활용도가 점차 확대되고 있다[1]. 그 중에서도 센서(Sensor)를 이용한 USN(Ubiquitous Sensor Network)은 센서노드와 리더기 간의 무선통신을 통해 사물과 사물간의 정보공유를 가능케 하는 핵심기술로서 관심을 끌고 있다[4].

USN을 활용한 다양한 연구 분야 중 사회적으로 가장 큰 이슈가 되고 있는 것이 u-Healthcare이다. u-Healthcare는 정보통신과 보건의료를 연결하여 언제 어디서나 예방, 진단, 치료, 사후 관리의 보건의료 서비스를 제공하는 것을 의미한다. 유·무선 네트워크를 바탕으로 환자, 의료기관, 솔루션 개발/기기업체 등의 유기적 연결을 통해 인간의 건강한 삶을 유지하기 위한 시스템이다[6].

본 논문에서는 u-Healthcare에 적합한 USN 미들웨어(Middleware) 시스템을 구현하여 이 기종의 센서 네트워크에서 획득한 데이터를 추상화하고 필터링, 이벤트 처리, 상황인식 처리를 통해 시스템의 상호 운용 및 안정성을 향상시켰다. 이는 어떠한 환경에서도 상호 운용성을 보장하기 위한 취지에서 진행되었다. 또한 본 논문에서 제시한 미들웨어 시스템을 이용하여 각 클러스터 센서노드의 정보와 사용여부 등을 미들웨어에 미리 저장하고, 데이터 전송 시 데이터를 감지해서 실시간 감시 및 조회가 가능하도록 지원하는 헬스 케어 시스템

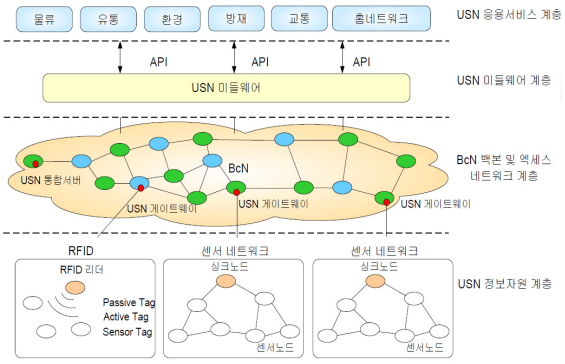
을 구현했다.

2. 관련 기술 및 연구

2.1 USN

USN은 유비쿼터스 사회를 구현하는 기본 인프라로 주목받는 핵심기술로 유통・물류・의료・환경을 비롯한 산업 및 생활 전반에 걸쳐 안전성과 편리함을 제공 할 수 있도록 센서 및 게이트웨이를 보급하여 센싱 및 네트워크 기능을 수행하는 시스템이다[5, 7].

USN 기술은 먼저 인식정보를 제공하는 전자태그를 중심으로 발전했고, 이에 센싱 기능이 추가되어 이들 간의 네트워크가 구축되는 과정을 통해 발전했다. 기존의 단순한 용도로만 쓰이던 전자태그를 모든 사물에 확대 적용할 수 있도록 기술이 발전하고 있으며, 더 나아가 단순 인식 기능뿐만 아니라 센싱 기능까지 지원하여 네트워크를 구축하는 자율 센서 망으로 발전했다. [그림 1]은 USN의 계층별 구조를 보여준다.



자료 : 김영만, "USN 기술개념 및 분류, 용어", 2006, 7.

[그림 1] USN의 계층 구조

2.2 USN 미들웨어

USN 미들웨어란 물리적으로 센서노드, 게이트웨이 등의 하드웨어와 응용 프로그램 사이에 존재하는 소프트웨어 계층으로 중간에서 하드웨어와 응용 프로그램의 통합이 유연하게 이루어지도록 지원한다.

또한 사용자에게 하부의 하드웨어, 운영체제, 네트워크에 상관없이 분산 컴퓨팅 환경, 원격 프로시듀어 콜, 메시징과 같은 서비스를 제공할 수 있도록 도와주는 소프트웨어이다[1].

현재 연구되고 있는 USN 미들웨어는 구성되는 위치에 따라 In-network 미들웨어와, Server-side 미들웨어로 구분이 가능하다[3].

본 논문에서는 두 가지 미들웨어 형태 중 논문의 특성에 맞게 Server-side 미들웨어를 설계하기 위해 기존에 연구된 TinyDB, DSWare, Cougar, MiLAN 등의 미들웨어 시스템들을 분석했다.

2.2.1 사례연구

TinyDB는 Berkeley 대학에서 연구한 데이터베이스 기반의 미들웨어로서 센서 네트워크를 가상의 분산 데이터베이스로 간주했다. TinyOS 기반으로 동작하는 미들웨어로써 Server와 in-Network aggregation 기능을 지원하며 자바 기반의 간단한 API를 지원한다. 그러나 TinyDB 는 TinyOS 기반에서 이용이 가능하다는 점의 단점이 있다[13].

DSWare(Data Service Middleware)는 Virginia 대학에서 연구한 데이터 서비스 미들웨어로 서비스들을 통합하여 제공하는 서비스 기반 미들웨어로써 응응 계층에서 필요한 이벤트 신청 및 탐지, 데이터 저장, 노드 관리 등 여러 서비스 응용 계층의 데이터 서비스를 추상화하여 제공한다. 하지만 특정 제품의 센서노드 하드웨어에 대한 의존적인 미들웨어로서, 이종의 센서노드들에 대한 추상적인 인터페이스 제공 기능이 부족하다[12].

Cougar는 Conell 대학의 데이터베이스 연구팀에서 연구하고 있는 센서 네트워크 기반의 분산 데이터 처리 시스템이다. Cougar에서 센서노드들의 데이터 접근과 처리는 모두 중앙 집중적이 아닌 분산된 형태로 처리된다. 데이터 처리 부분에서는 효율적 이지만 지역적 정보에 의존해야 하는 문제가 발생한다[16].

Milan(Middleware Linking Application and Network)은 Rochester 대학에서 스마트 메디컬 홈을 위해 개발한 USN 미들웨어이다. Milan은 센서 네트워크의 수명은 최대화 하면서 서비스 품질에 대한 요구를 최대로 만족시키기 위해 서비스 품질 요구 처리기능을 지원한다. 하지만 USN 응용 서비스에 tightly-coupled 되어 있어서 이 기종의 센서노드들에 대한 추상화를 지원하지 않고 모바일

센서노드를 지원하지 않는 단점이 있다[8].

2.2.2 한계점 및 연구방향

앞에서 설명한 바와 같이 현재 진행되고 있는 미들웨어는 어플리케이션의 다양한 질의 수용, 자바를 통한 간단한 API 인터페이스 기능 지원, 응용 계층으로 제공하는 이벤트 신청 및 탐지, 데이터 저장, 노드 및 노드 그룹을 관리하는 서비스 추상화 등의 다양한 서비스를 제공한다. 그러나 대부분의 미들웨어들은 특정 하드웨어 및 운영체제서만 이용이 가능한 의존적 단점과 이 기종간의 센서노드에 대한 추상화 기능을 제공하지 않는 한계점을 갖고 있다.

따라서 본 논문에서는 하드웨어 및 운영체제에 의존적이지 않고, 이 기종의 센서노드에서 획득한 데이터에 대한 추상화 기능을 지원하며 획득한 데이터에 대한 상황인식 및 이벤트 관리, 연속적인 중복 데이터의 무시와 다양한 질의에 대한 처리를 지원하는 미들웨어를 설계했다.

<표 1> 미들웨어 요구사항 분석

구분	상세내용	처리 시점	비고
미들웨어 요구사항	● 데이터 추상화 - 이 기종 센서노드 정보 미리 저장 - 사용여부, 시리얼번호, DB접속정보 설정 - 데이터 추상화 후 데이터 전달	데이터 수신	
	● 데이터 필터링 - 입력되는 데이터 값을 비교 - 전 단계와 같은 값의 데이터는 DB에 미 저장	데이터 수신	
	● 상황인식 서비스 - 상황인식이 필요한 데이터 값 설정 - 입력되는 데이터 값이 설정된 범위에 해당 - 상황인식 서비스 실시	상황 인식 범위 해당	
	● 이벤트 서비스 - 이벤트가 필요한 데이터 값 설정 - 입력되는 데이터 값이 설정된 범위에 해당 - 이벤트 서비스 실시	이벤트 범위 해당	

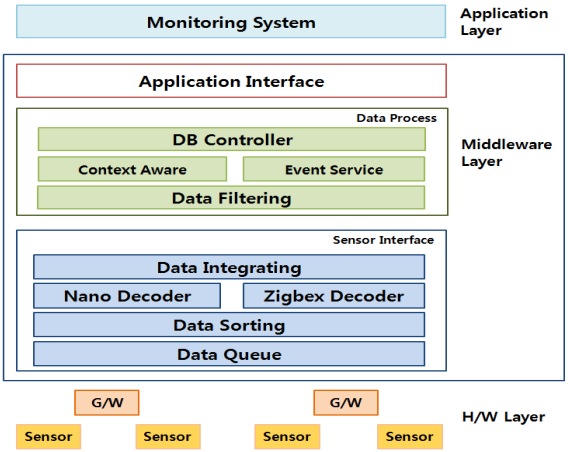
3. 미들웨어 설계 및 알고리즘

3.1 미들웨어 요구사항

본 논문에서 분석한 미들웨어 요구사항은 기능별 필요한 위치에 따라 크게 센서 인터페이스, 데이터 프로세스, 응용 인터페이스의 3가지로 분류할 수 있다. 센서 인터페이스에서는 데이터 추상화 기능을 수행하고 데이터 프로세스에서는 데이터 필터링, 상황인식 서비스, 이벤트 서비스를 담당한다. 또한 응용 인터페이스에서는 질의 처리를 통한 외부와의 연결을 지원한다.

3.2 미들웨어 구성

[그림 2]에서 제시한 u-Healthcare 시스템은 크게 센서노드와 게이트웨이를 포함한 H/W Layer와 데이터의 통합, 처리, 저장을 담당하는 Middleware Layer, 모니터링을 지원하는 Application Layer로 구분한다. 본 논문에서 설계한 Middleware Layer는 Sensor Interface, Data Process, Application Interface 컴포넌트로 구성했다.



[그림 2] 미들웨어 계층도

Sensor Interface 컴포넌트에서는 데이터를 순차적으로 입력받는 Data Queue와, 상이한 형태의 데이터를 구별하는 Data Sorting, 각 형태별 데이터 형식을 공통형식으로 변환하는 Nano Decoder와 Zigbex Decoder 그리고 공통형식의 데이터를 통합하는 Data Integrating 모듈로 구성된다.

Data Process 컴포넌트에서는 연속적 중복되는 데이터를 처리하는 Data Filtering, 이벤트 서비스를 지원하는 Event Service, 상황인식을 지원하는 Context Aware, DB에 데이터를 저장하는 DB Controller 모듈로 구성된다.

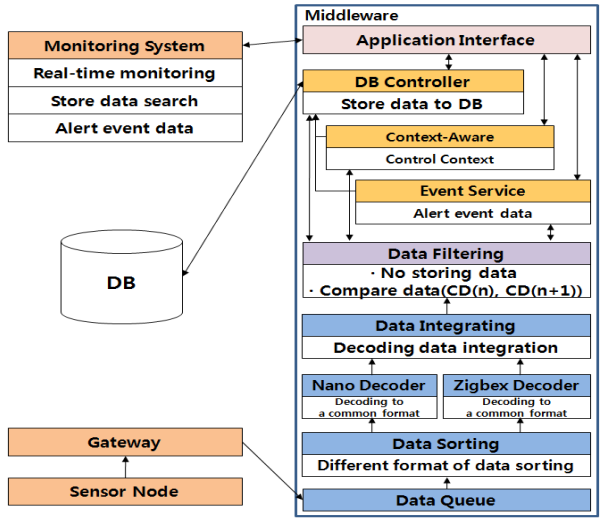
Application Interface 모듈에서는 Monitoring

System과 연계하여 데이터를 Display하고 사용자 질의를 처리한다.

3.2.1 미들웨어 구조

[그림 3]은 본 논문에서 설계한 미들웨어의 모듈별 흐름과 동작을 보여주고 있다. 게이트웨이를 통해 입력된 데이터는 Data Queue 모듈을 통해 순차적으로 입력되고 Data Sorting 모듈로 전달된다.

Data Sorting 모듈에서는 상이한 데이터 형식에 따라 구분해서 Nano Decoder 모듈과 Zigbex Decoder 모듈로 각각 전달한다. Nano Decoder 모듈과 Zigbex Decoder 모듈에서는 전송된 형태의 데이터를 지정된 공통형식으로 변환한다. 이렇게 변환된 데이터는 각각 Data Integrating 모듈로 전달되어 순차적으로 통합된다.



[그림 3] 미들웨어 모듈별 흐름도

이렇게 추상화 과정을 거쳐 공통형식으로 변환된 데이터는 Data Filtering 모듈로 전달되어 연속적으로 입력된 데이터를 전 단계의 데이터와 비교해서 동일 데이터에 대해서는 DB에 저장기능을 제공하지 않는다. 또한 Data Filtering에서는 입력된 데이터 값에 대해서 Application에서 미리 지정한 Context-Aware와 Event Service 값의 범위에 해당하는 경우 데이터를 각각 Context-Aware와 Event Service 모듈로 전달한다.

위 두 가지를 모두 만족시키지 않는 경우는 DB Controller 모듈로 전달해서 DB에 저장하도록 한다. Event Service 모듈에서는 Warning Sign을 통해 불의의 사고를 미연에 방지하고 Context-Aware 모듈에서는 냉·난방기 가동 등의 상황에 맞는 서비스를 제공한다. DB Controller 모듈은 데이터 값과 Event 수행여부를 DB에 저장한다.

Application Interface 모듈에서는 Monitoring System과 연계하여 데이터를 Display하고 사용자 질의를 처리한다.

Middleware Algorithm
  Repeat
    Receive data $D_a(n)$, $D_b(n)$ from Gateway
    Send $D_a(n)$, $D_b(n)$ to Data Queue

  Data Sorting Algorithm:
    Receive $D_a(n)$, $D_b(n)$ from Data Queue
      $D_a(n)$, $D_b(n)$ Sorting to Nano, Zigbex
    Send $D_a(n)$, $D_b(n)$ to Nano Decoder, Zigbex Decoder
    //데이터를 Nano, Zigbex로 구분, 각각 전달

  Nano Decoder Algorithm:
    Receive $D_a(n)$ from Data Sorting
      $D_a(n)$ decode to $CD(n)$ Send $CD(n)$ to DI
    //Nano $D_a(n)$을 공통형식 $CD(n)$으로 변환

  Zigbex Decoder Algorithm:
    Receive $D_b(n)$ from Data sorting
      $D_b(n)$ decode to $CD(n)$ Send $CD(n)$ to DI
    //Zigbex $D_b(n)$을 공통형식 $CD(n)$으로 변환

  Data Integrating Algorithm:
    Recive $CD(n)$, $CD(n)$
      from Nano Decoder, Zigbex Decoder
    Send $CD(n)$, $CD(n+1)$ to Data Filtering
    //두 $CD(n)$을 $CD(n)$, $CD(n+1)$로 통합

  Data Filtering Algorithm:
    Receive $CD(n)$, $CD(n+1)$ from Data Integrating
      Storing $CD(n)$
      if ($CD(n+1)$ == $CD(n)$)
        no storing $CD(n+1)$
    // 전 단계와 비교 후 일치할 경우 no storing
      else if $CD(n+1)$ == ES)
        Send $CD(n+1)$ to Event Service
      else Send $CD(n+1)$ to DB Controller
    // ES에 해당하면 Event Service로 전송
    // 나머지 경우 DB Controller로 전송

Context Aware Algorithm:
    Receive $CD(n+1)$ from Data Filtering
      A_Context()
    Send $CD(n+1)$ to DB Controller

  Event Service Algorithm:
    Receive $CD(n+1)$ from Data Filtering
      A_event()
    Send $CD(n+1)$ to DB Controller

  DB Controller Algorithm:
    Receive $CD(n+1)$ from Data Filtering
      Store $CD(n+1)$

[그림 4] 미들웨어 알고리즘

3.3 미들웨어 알고리즘

[그림 4]는 구현된 미들웨어에서 데이터를 통합, 처리하는 알고리즘으로 게이트웨이로부터 넘겨받은 데이터를 미들웨어에서 처리해서 DB에 저장하는 과정을 나타내고 있다.

이 기종의 센서 네트워크에서 획득한 데이터 $D_a(n)$과 $D_b(n)$은 Data Sorting Module 에서 각각 Nano, Zigbex형태로 구분되어 각각 Nano Decoder와 Zigbex Decoder로 전송된다. 각 Decoder에서는 $D_a(n)$과 $D_b(n)$을 공통형식 $CD(n)$으로 변환하고 Data Integrating Module로 보내 순서에 맞게 $CD(n)$과 $CD(n+1)$로 통합한다.

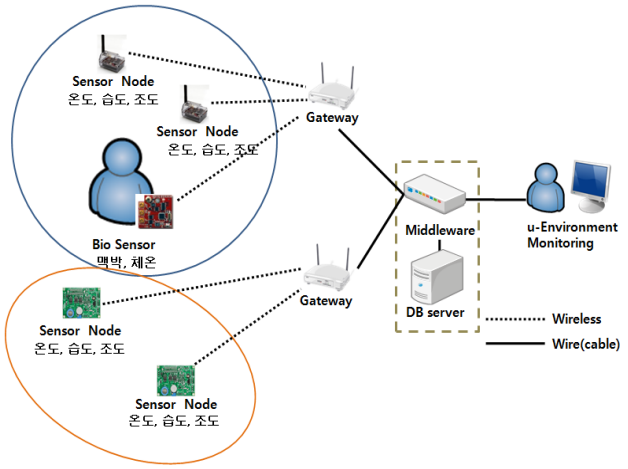
공통 형식으로 통합된 데이터는 Data Filtering Module을 통해 연속적으로 입력되는 $CD(n)$과 $CD(n+1)$을 비교해서 중복되는 데이터의 경우는 DB Controller Module로 보내지 않아서 저장되지 않도록 한다. 또한 Context-Aware와 Event Service Module에서 미리 설정한 데이터 범위에 일치하는 데이터가 전송될 경우, 각각 Context-Aware와 Event Service로 전송해 해당 액션을 취하게 한다.

나머지 값에 대해서는 DB Controller Module로 전송해 DB에 저장한다.

4. u-Healthcare 시스템 구현

4.1 시스템 구성

[그림 5]는 본 논문에서 구현한 헬스 케어 시스템 전체 구성도를 보여주고 있다.



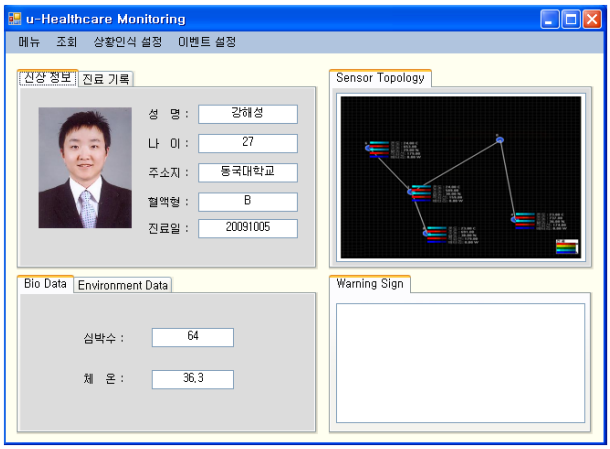
[그림 5] 본 논문에서 구현한 헬스 케어 시스템

Nano-Qplus 환경에서 포팅 되어 온도, 습도, 조도 등을 측정하는 Sensor Node 2개와 TinyOS에서 포팅 되어 온도, 습도, 조도를 측정하는 센서노드 2개, 그리고 심박수와 체온을 측정하는 BIO센서 1개를 사용했다.

OS 별로 Nano-Qplus와 TinyOS의 두 개의 클러스터로 구성되며 각각 상이한 형태의 센서 네트워크를 통합하기 위해 각 센서의 포트 번호, DB 접속 정보 등을 미리 입력해서 게이트웨이를 통해 전달된 데이터를 미들웨어가 인식할 수 있게 했다. 획득한 데이터는 미들웨어를 통해 상이한 메시지 형태를 표준 공통 형식으로 변환해서 상황에 맞게 처리한다.

4.2 시스템 구현

[그림 6]은 본 논문에서 구현한 미들웨어를 적용한 헬스 케어 모니터링 화면이다. 메뉴구성을 통해 환자조회가 가능하고 상황인식과 이벤트 서비스 범위 설정이 가능하다. 또한 환자조회를 통해 신상정보를 확인할 수 있다. 또한 BIO Sensor를 통해 획득한 심박 수와 체온의 값을 모니터링 할 수 있으며, 이 기종 센서 네트워크에서 획득한 데이터를 추상화해서 하나의 Topology View에 나타냈다. 그리고 Warning Sign 메시지를 통해 응급상황에 대처할 수 있게 구현했다.



[그림 6] 헬스 케어 모니터링 화면

5. 결론

본 논문에서는 기존 미들웨어의 문제점으로 인

식되고 있는 특정 하드웨어에 종속적이라는 단점을 보완하기 위해 이 기종의 센서 네트워크에서 획득하는 데이터에 대한 추상화 기능을 지원했다.

또한 연속적으로 반복되는 데이터의 저장을 줄임으로 인해 서버 부하를 감소시키고 설정된 범위에 맞는 상황인식 및 이벤트 처리를 통해 점점 지능화되는 미들웨어를 설계했다.

설계된 미들웨어를 사회적 이슈가 되고 있는 IT 의료분야에 적용해서 헬스 케어 모니터링 시스템을 구현했다. 센서를 통해 온도, 습도, 조도, 심박수, 체온을 측정해서 실시간으로 관리하고 모니터링 하는 기능을 구현했다.

본 논문에서 구현한 미들웨어 기반의 헬스 케어 모니터링 시스템을 통해 급변하는 환경에서 USN을 활용한 보건 의료 관리의 가능성을 확인할 수 있었다.

향후 더 많은 센서 네트워크를 추상화하고 이기종의 하드웨어를 직접 제어하는 확장된 개념의 추상화를 가능하게 하는 연구와 상이한 환경의 미들웨어를 통합하는 통합 미들웨어에 대한 연구도 함께 진행되어야 할 것이다.

[참고문헌]

[1] 김민수, 이용준, “USN 미들웨어 기술 개발 동향”, 전자통신 동향분석, 22권, 3호, pp.67-79, 2007.

[2] 김영만, "USN 기술개념 및 분류, 용어", pp.1~65, 2006, 7.

[3] 김영만, 한재일, “센서 미들웨어 기술”, 정보과학회지, 25권, 12호, pp.35~48, 2007. 12.

[4] 김의창, “A Simulation of middleware for the u-Healthcare System”, International Conference on Korea-Chinese Collaboration for Global e-Business, GeBA, Beijing, China, June, pp.285-295, 2008.

[5] 정부만, "2006년도 국내외 USN 산업동향 분석 연구“, 한국정보사회진흥원, 연구개발 결과보고서, 2006. 9.

[6] 지경용 외, “유비쿼터스 시대의 보건의료”, 진한M&HB, 2006.

[7] 진희채, “공공부문 USN 도입 방안에 과난 연구”, 정보사회진흥원, 연구개발 결과보고서, 2006.

[8] A. Murphy and W. Heinzelman, "MiLAN: Middleware Linking Applications and Networks", TR-795, University of Rochester, Computer Science, pp.1~14, 2003. 1.

[9] C. Shen, C. Srisathapornphat, C. Jaikeo, “Sensor Information Networking Architecture and Applications,” IEEE Personal Communications, Vol.8, No.4, pp.52-59, 2001. 8.

[10] D.H. Wilson, A.C. Long, C. Atkeson, “A Context -Aware Recognition Survey for Data Collection Using Ubiquitous Sensors in the Home”, In Proceedings of CHI 2005: Late Breaking Results, pp.1865-1868, 2005. 4.

[11] H. Liberman and T. Selker, “Out of Context : Computer System That Adapts to, and learn from, context", IBM Systems Journal, Vol.39, No.3&4, pp.617~632. 2000.

[12] S. Li, S. Son, and J. Stankovic, "Event Detection Services Using Data Service Middleware in Distributed Sensor Networks", Int'l Workshop Information Processing in Sensor Networks(ISPN'03), Palo Alto, CA, pp.502~517, 2003.

[13] S. R. Madden, M. J. Franklin, and J. M. Hellerstein, "TinyDB: An Acquisitional Query Processing System for Sensor System Networks", ACM Trans. Database Systems, Vol. 30, No. 1, pp.122~173, 2005.

[14] Salem, H., and Nader, M., "Middleware Challenges and Approaches for Wireless sensor Network", IEEE Distributed System Online, Vol.7, No.3, pp.1~23, 2006. 3.

[15] W. Rabiner Heinzelman, A. Chandrakasan, and H.Balakrishnan, “Energy-Efficient Communication Protocol for Wireless Micro- sensor Networks", Proc. of the 3‘rd Inter national Conf. on System Sciences (HICSS'00), pp.1~10, 2000. 1.

[16] Y. Yao and J. Gehrke, "The Cougar Approach to In Network Query Processing in Sensor Networks", SIGMOD Record, Vol.31, No.3, pp. 9~18, 2002. 10.