무선 센서 네트워크를 이용한 홈 네트워크 개인화 서비스

나선웅° 오홍록 이해각 이상정
순천향대학교 컴퓨터학부
{nsw97°, sarision, lhk7083, sjlee }@sch.ac.kr

Personalized Home Network Service using Wireless Sensor Network

Sun-Wung Na°, Hong-Rok Oh, Hae-Kag Lee, Sang-Jeong Lee
Dept. of Computer Science and Engineering, Soonchunhyang University

요 약

본 논문에서는 홈 네트워크 환경에서 댁내에 배치된 센서 노드들과 사용자에게 부착된 센서 노드로 구성된 무선 센서 네트워크를 이용하여 개인화 서비스를 제공하는 서비스 모델을 제안한다. 제안된 서비스 모델은 홈 서버에 등록된 사용자 프로파일과 센서 노드들로부터 수집된 상황정보를 이용하여 사용자 개인의 특성에 따라 댁내 가전기기들을 설정하고 연동 서비스를 제공한다. 제안된 방식은 TinyOS가 탑재된 Moteiv사의 tmote sky 모트들을 사용하여 구현 테스트한다.

1. 서 론

최근 댁내의 가전기기들을 유무선 네트워크에 연결하여 지능화된 서비스 제공하는 지능형 홈 네트워크에 많은 관심과 연구개발이 진행되고 있다. 지능형 홈 네트워크 구성을 위해서는 사용자의 댁내 기기에 대한 간섭을 최소화하면서 사용자 개인의 특성에 맞추어진 서비스를 제공해야 한다. 이를 위해서는 댁내 사용자의 주변의 상황정보의 실시간 수집이 필수적이다. 또한 무선 및 프로세서 기술의 발달로 작고 저전력의 무선 센서 노드들로 구성된 센서 네트워크를 구성하여 상황 데이터를 수집하여 활용하고자 하는 연구가 활발히 진행되고 있다.

본 논문에서는 홈 네트워크 환경에서 댁내에 배치된 센서 노드들과 사용자에게 부착된 센서 노드로 구성된 무선 센서 네트워크를 이용하여 개인화 서비스를 제공하는 서비스 모델을 제안한다. 제안된 서비스 모델은 홈 서버에 등록된 사용자 프로파일과 센서 노드들로부터 수집된 상황정보를 이용하여 사용자 개인의 특성에 따라 댁내 가전기기들을 설정하고 연동 서비스를 제공한다. 또한 홈 서버에 사용자의 프로파일 데이터를 등록하고 관리하는 프로파일 관리기를 개발하고 이를 이용하여 사용자의 상황에 적합하게 기기들을 동작시켜 개인화된 서비스를 제공한다. 제안된 방식은 TinyOS[1]가 탑재된 Moteiv 사의 tmote sky 모트[2]들을 사용하여 구현 테스트한다.

2. 무선 센서 네트워크

이 논문은 2005 년도 중소기업청에서 지원하는 기술연구회 공동 연구개발사업에 의하여 연구되었음(S0505616-J1530003-15000011)

홈 네트워크 환경에서 사용자 상황에 맞는 개인화 서비스를 제공하기 위해서는 개인 프로파일이 제공되어야 하며, 개인 프로파일 데이터를 작성하기 위해서는 댁내의 사용자 환경 데이터가 필요하다. 댁내에 배치된 센서 네트워크는 온도, 조도, 습도, 압력, 소리 등의 환경데이터를 수집하여 홈 서버에 전송하고, 홈 서버는 수신된 환경 데이터를 사용하여 사용자의 특성에 맞추어진 냉난방, 조도조절 등을 위해 기기들을 제어한다. 무선 센서 노드들은 저전력을 소비하면서 크기가 작기 때문에 댁내의 보이지 않는 곳에 설치가 가능하다. 또한 무선통신으로 데이터를 주고 받기 때문에 새로운 유선망을 설치할 필요가 없다는 이점도 있다.

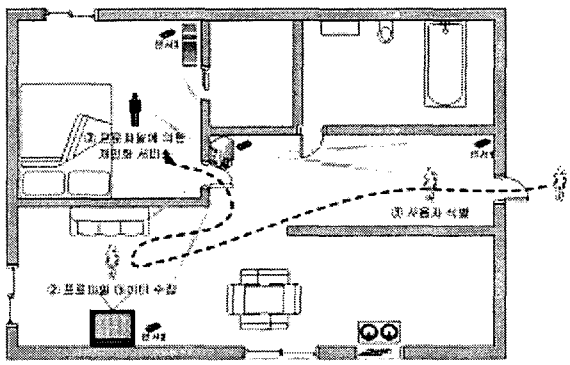

그림 1 무선 센서 네트워크 시나리오 예

그림 1은 홈 네트워크환경에서 무선 센서네트워크를 이용한 개인화 서비스 모델의 시나리오 예를 나타낸다.

사용자가 댁외에서 들어와 거실에서 TV 시청 후 방안으로 들어가는 과정 중에 일어나는 개인화 서비스 예를 보여주고 있다. 사용자는 고유의 ID를 부여한 센서를 지니고 있으며, 프로파일 관리자에 등록된 가전기기 근처에는 환경 데이터를 수집할 수 있는 고정 센서 노드가 설치 되어있다. 사용자가 외출하거나 댁내에 들어왔을 때 사용자가 가지고 있는 센서의 고유 ID를 수신 받아 사용자를 식별한다(①사용자 식별). 이 때 사용자의 외출시간, 퇴근시간이 홈 서버의 프로파일 관리자에 저장된다. 홈 서버는 댁내에 사용자가 없다고 판단되면 댁내 기기들을 외출모드로 설정하고 방범 기능을 작동시킨다. ②와 같이 사용자가 프로파일에 등록된 가전기기를 작동하면 어떤 사용자가 어떤 시간대에 기기를 어떻게 작동하는지와 그때의 디바이스 근처에 위치한 고정 센서노드를 통해 온도, 습도, 조도를 포함하는 환경 데이터가 수집된다. ③은 사용자가 홈 서버의 프로파일 관리자에 설정된 사용자의 프로파일 데이터를 통해 홈 서버로부터 설정조건에 일치하는 경우 사용자에게 맞는 개인화 서비스를 제공함을 보여준다.

3. 개인화 서비스

홈 네트워크 환경에서 사용자에게 상황 인식 개인화 서비스를 제공하기 위해서는 사용자의 정보, 사용자의 선호도, 디바이스 설정, 디바이스 사용 권한 등을 포함하는 데이터가 필요하다. 이러한 데이터를 수집하고 관리하기 위해 사용자 프로파일을 정의하고 프로파일 관리자를 구현하였다.

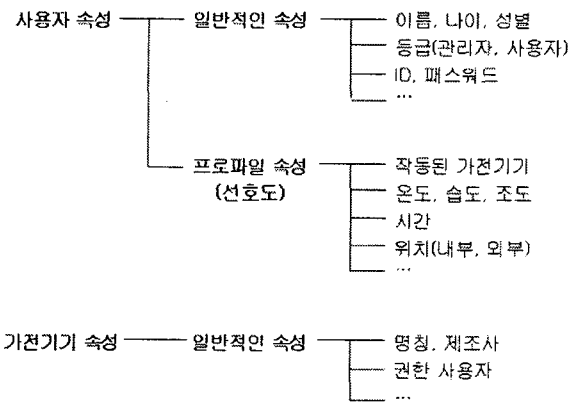

그림 2 프로파일 구성요소

그림 2 는 본 논문에서의 상황 정보를 위해 정의한 프로파일 구성요소이다. 구성요소는 크게 사용자 속성과 디바이스 속성으로 나누어지며 상황인식 서비스에 적용된다. 사용자의 일반적 속성은 사용자의 이름, 나이, 성별 등의 기본정보, 사용자 등급, 가전기기 초기 설정 시 인증을 위한 ID 와 패스워드 요소로 구성된다. 프로파일 속성은 사용자가 기기를 작동했을 때 수집되는 요소들로서 작동된 디바이스, 시간, 센서로부터 수집되는 온도, 조도, 습도, 사용자 위치 등이며 사용자의 개인화 서비스 제공에 사용된다. 가전기기 속성은 기기에 대한 명칭, 제조사 등의 일반적인 요소와 기기에 대한 사용자의 권한을 나타내는 권한 사용자 요소로 구성된다. 홈 네트워크 환경 사용자는 관리자와 일반 사용자로 구분되어 정의된다. 관리자는 일반사용자의 프로파일을 생성 및 수정 할 뿐 아니라 서비스(기기동작)에 대한 접근 권한을 부여 할 수 있다. 일반사용자는 홈 네트워크 환경에서 자신에게 부여된 접근 권한에 따라 서비스를 이용한다. 사용자 프로파일의 등록 및 관리는 홈 네트워크의 홈 서버에 구현된 프로파일 관리기 (profile manager)에 의해 관리된다. 관리기는 PDA 나 WAP 을 이용한 휴대폰을 이용해 프로파일 관리기에 무선으로 접근할 수 있고, 웹 브라우저 또는 집안 내에 있는 월패드(wall pad)를 통해 접근할 수 있다. 프로파일 관리기에 접근하게 되면 패스워드로 인증과정을 거치게 된다.

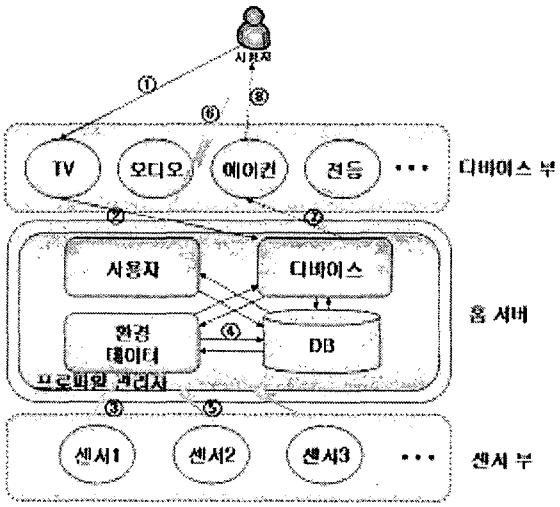

그림 3 프로파일 저장 및 적용에 대한 구성도

그림 3은 프로파일 저장 및 적용에 대한 동작 구성도를 나타낸다. 사용자의 프로파일 데이터 수집 과정은 ①사용자가 프로파일 관리기에 등록된 가전기기를 작동시켰을 때 ②작동된 디바이스는 ON,OFF 여부를 프로파일 관리기의 가전기기 관리자에게 알려준다. ③프로파일 관리기는 사용자에 부착된 센서로부터 사용자의 정보와 작동된 기기 주변에 있는 고정 센서로부터 온도, 조도, 습도 등의 환경데이터를 수집한다. 그리고 ④사용자 정보와 가전기기 정보, 동작된 기기 주변의 환경데이터를 프로파일화 하여 데이터베이스에 저장하게 된다. 프로파일에 의한 개인화 서비스 동작은 사용자가 사전에 설정한 시간 대의 온도보다 현재 주변 온도가 높은 경우를 가정하여 에어컨을 ON 시키는 과정에서 ⑤홈 네트워크 환경 내에 설치된 고정 센서로부터 온도 값을 센싱하여 비교하고, ⑥사용자 센서로부터 사용자가 집안 내부에 있는지를 판단하여 조건에 맞을 경우 ⑦기기들 동작시켜 ⑧사용자에게 개인화된 서비스를 제공하게 된다.

4. 구현 및 테스트

제안된 방식은 TinyOS[1]가 탑재된 Moteiv사의 tmote sky 모트(TelosB)[2]들을 센서노드로 사용하여 무선 센서 네트워크를 구축하고, 윈도우 기반의 홈 서버 상에서 프로파일 관리자를 VC++로 구현 하였다. 센서노드는 무선통신을 위해 ZigBee 프로토콜을 이용하며, 250Kbps의 처리량, 10KB RAM , USB 인터페이스를 가지고 있다. 센서 수집 및 무선통신은 네트워크 임베디드 시스템을 위한 언어인 nesC 언어[3]로 구현하였다.

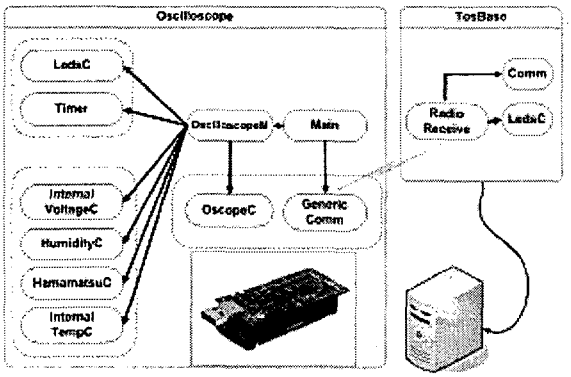

그림 4 센서 노드 프로그램의 컴포넌트 구성도

그림 4는 본 논문 구현에 쓰인 센서 노드의 프로그램 컴포넌트 구성도를 나타내는 그림이다. 원은 프로그램의 주요 컴포넌트를 표시하고 에지는 인터페이스 관계를 표시한다. Oscilloscope 프로그램은 센서 노드에서 온도, 조도, 습도, 내부 전압, 내부 온도등을 수집하여 패킷화한 데이터를 전송하는 프로그램이며 TosBase 프로그램은 패킷을 수신하면 Group ID 를 확인하고 자신의 ID 와 같으면 받아서 시리얼로 전송하는 단순히 브리지 역할을 하는 프로그램이다. Oscilloscope의 컴포넌트들은 센서 노드의 하드웨어적 동작을 도와주는 부분과 데이터를 수집하는 부분, 수집한 데이터를 패킷화하는 메인부분, 패킷화된 데이터를 전송하기 위한 부분으로 분류했다. 센서 노드에 있는 센서를 통해 수집된 데이터는 패킷화되고 Zigbee 프로토콜을 통해 전송되며, PC 에 연결된 TosBase 프로그램이 업로드된 모트는 전송된 패킷을 수신한다. 전송되는 패킷은 목적지 주소, 그룹 ID, 메시지 길이, 소스 주소, 채널, 데이터 등으로 구성된다. 메시지의 원본 데이터는 홈 서버의 프로파일 관리기에서 해석하여 본 논문에서 사용되는 환경 데이터인 온도, 조도, 습도 등의 값으로 변환된다.

```
event result_t Timer.fired() {
  call Timer.start(TIMER_ONE_SHOT, 100);
  switch(state) {
  case HUMIDITY:
    call Humidity.getData();
    break;
  case TEMPERATURE
    call Temperature.getData();
    break;
  case TSRSENSOR:
    call TSR.getData();
    break;
  case PARSENSOR:
    call PAR.getData();
    break;
  case ITEMP:
    call InternalTemperature.getData();
    break;
  case VOLT:
    call InternalVoltage.getData();
    break;
  default:
    call Timer.start(TIMER_ONE_SHOT, 10);
  }
  return SUCCESS;
}
```

```
OscilloscopeM.Timer -> TimerC.Timer[unique("Timer")];
OscilloscopeM.Leds -> LedsC;
OscilloscopeM.HumidityControl -> HumidityC;
OscilloscopeM.Humidity -> HumidityC.Humidity;
OscilloscopeM.Temperature -> HumidityC.Temperature;
OscilloscopeM.TSR -> HamamatsuC.TSR;
OscilloscopeM.PAR -> HamamatsuC.PAR;
OscilloscopeM.InternalTemperature -> InternalTempC;
OscilloscopeM.InternalVoltage -> InternalVoltageC;
OscilloscopeM.HumidityError -> HumidityC.HumidityError;
OscilloscopeM.TemperatureError -> HumidityC.TemperatureError;
OscilloscopeM.OHumidity -> OscopeC.Oscope[0];
OscilloscopeM.OTemperature -> OscopeC.Oscope[1];
OscilloscopeM.OTSR -> OscopeC.Oscope[2];
OscilloscopeM.OPAR -> OscopeC.Oscope[3];
OscilloscopeM.OInternalTemperature -> OscopeC.Oscope[4];
OscilloscopeM.OInternalVoltage -> OscopeC.Oscope[5];
```

그림 5 프로그램의 주요 소스

그림 5 는 Oscilloscope 프로그램 중 주요 소스 부분들이다. 왼쪽 그림은 모듈에서 이벤트를 호출하면 100ms 마다 각 센서들마다 순회하여 getData()함수를 통해 데이터를 수집하는 것을 보여주며 오른쪽 그림은 각 센서들을 통해 수집된 데이터들을 정해진 채널별로 전송되는 것을 보여준다. 라운드 로빈 방식으로 데이터가 준비된 채널을 순회하며 TOS 메시지의 페이로드에 센서 데이터 10 개를 복사한 후 Oscilloscope 패킷 형태로 만들고 sendHelper() 호출한다. sendHelper()는 Gerneric Comm 모듈을 통해 DataMsg.send() 호출하여 데이터 송신한다.

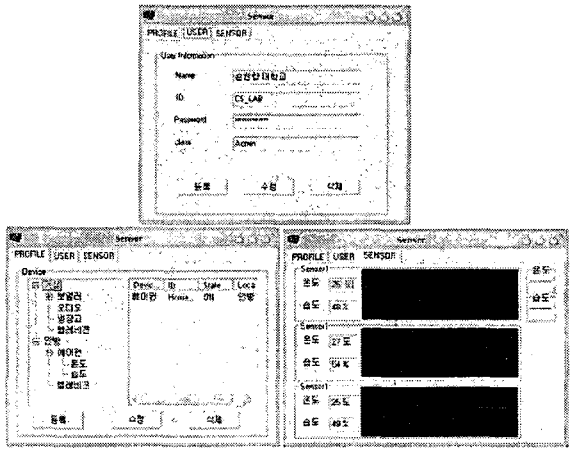

그림 6 사용자 및 디바이스 등록, 센서 모니터링 GUI

그림 6 은 본 논문에서 구현한 GUI 로서 사용자 등록, 수정, 삭제 및 디바이스 등록, 수정, 삭제 부분과 각 센서로부터 수집되는 데이터를 실시간 모니터링 할 수 있는 화면으로 구성하였다.

6. 결론

본 논문에서는 홈 네트워크 환경에서 댁내에 배치된 센서 노드들과 사용자에게 부착된 센서 노드로 구성된 무선 센서 네트워크를 이용하여 개인화 서비스를 제공하는 서비스를 개발 제안하였다. 향후 센서 네트워크를 사용하여 댁내에서 사용자의 정확한 위치를 파악하고, 프로파일 정보를 이용하여 사용자의 선호도를 예측하는 방식을 추가할 예정이다.

7. 참고문헌

[1] TinyOS, http://www.tinyos.net/
[2] Moteiv사, http://www.moteiv.com
[3] nesC, http://nescc.sourceforge.net/
[4] Raúl Jimeno et.al " An architecture for the personalized control of domotic resources" , Proceedings of the 2nd European Symposium on Ambient Intelligence,pp. 51-53, 2004.