

RapidPLUS를 이용한 센서노드 시물레이션

안성순*, 이정배**, 한형욱*, 김대웅**

*선문대학교 전자계산학과

**선문대학교 컴퓨터정보학과

e-mail: ssAhn83@gmail.com, jblee@sunmoon.ac.kr, silencex@nate.com

kde6767@dreamwiz.com

A RapidPLUS used Sensor Node Simulation

SungSoon Ahn*, Jeong-Bae Lee**, Hyeong-Wook Han*, Dae-Eung Kim**

*Dept of Computer and Information Science, SunMoon Univ.

요 약

유비쿼터스 컴퓨팅은 우리의 생활양식을 빠르게 변화시켜가고 있다. 그러한 배경에는 IT 기술 고도화와 무선 통신기술 발달로 인한 센서 네트워크 환경이 있다. 유비쿼터스 환경 속 센서 네트워크 구축에 있어서는 기존 네트워크와는 다르게 정해진 규격의 동일 장비의 네트워크가 아닌 정해진 규격의 다양한 장비들 간의 상호교류 속에서 이루어진 다는 것이 다르다. 또한 무선을 통해 정보를 주고 받기 때문에 주변 환경에 따라서 통신을 위한 추가적인 경비가 더 소모되기도 한다. 본 논문에서는 이러한 센서 네트워크 환경 구축함에 있어 효율적인 방안을 제시하고 RapidPLUS를 이용하여 시물레이터를 구현한다.

1. 서론

유비쿼터스 컴퓨팅[1-2]은 기존에 사용자가 서비스를 요청하기 위해 정해진 시간에 직접 컴퓨터가 있는 곳으로 가서 일정한 방법에 의해 요청을 하고 그 요청에 따른 결과값을 기다리는 것과는 다르게 사용자가 어떤 시간이나 장소에 구애 받지 않고 네트워크를 통해 서비스를 요청하고 네트워크로 서비스를 받는 것을 이야기하는데, 이러한 환경은 고도화된 IT기술과 네트워크 기술이 선결조건이라 할 수 있다. 그러한 기술들은 최근 급속히 발전하였고, 많은 유비쿼터스 제품들이 출시가 되고 있으며, 우리는 그러한 제품들을 생활 속에서 쉽게 접하고 있다.

그 중에서 유비쿼터스 센서 네트워크는 센서노드를 통하여 물리공간의 빛, 소리, 온도, 움직임등을 감지하고 측정하여 사용자가 요청하기 전에 사용자가 원하는 서비스를 제공해준다. 그러한 서비스는 최근 “화재예방시스템”, “방범시스템”등과 같은 무인 관리 시스템에 적용되어 활용되어지고 있다. 하지만 센서 노드가 가지고 있는 센싱의 범위나 통신의 상태는 사람의 눈으로 직접적으로 볼 수 없기 때문에 어떻게 센싱되고 동작하는지 직관적으로 파악할 수 없다. 그래서 센서 노드를 개발하거나 네트워크를 구축함에 있어서는 외부 환경 요인 또는 사용자 요구사항에 따라서 센서 노드의 재개발 혹은 센서 노드의 추가 생산이 불가피하여 추가적인 인력이나 경비가 소모될 뿐 아니라 시장의 진입시점도 늦어진다.

그래서 센서 노드를 개발하거나 네트워크를 구축하기 이전에 노드가 가진 정보들과 타겟이 되는 주변환경의 정보를 S/W적으로 프로토타이핑을 하여 사전에 노드를 배치

하여본다면 앞에서 언급한 문제점은 줄어들 것이고, 그것은 곧 시간, 인력, 경비의 절감으로 이어지며 시장에 대한 진입 속도 또한 빠르게 될 것이다.

본 논문에서는 기존 도시나 주택을 가상으로 시물레이션 프로그램을 소개하고, 센서 노드 배치에 대한 시물레이터를 RapidPLUS를 사용하여 구현한다.

2. 생활공간 시물레이션

센서 노드는 특정한 구역을 만들고 그 만들어진 공간에 노드를 배치하여 네트워크를 구축하기도 하지만, 대부분은 기존에 가지고 있는 생활공간 내에서 센서 노드의 배치를 통하여 네트워크를 구축한다. 그렇기 때문에 센서 노드가 동작 가능한 주변 환경인지 알 수 있어야 한다. 그러한 주변환경을 가상으로 시물레이션 가능한 S/W를 본 장에서는 소개한다.

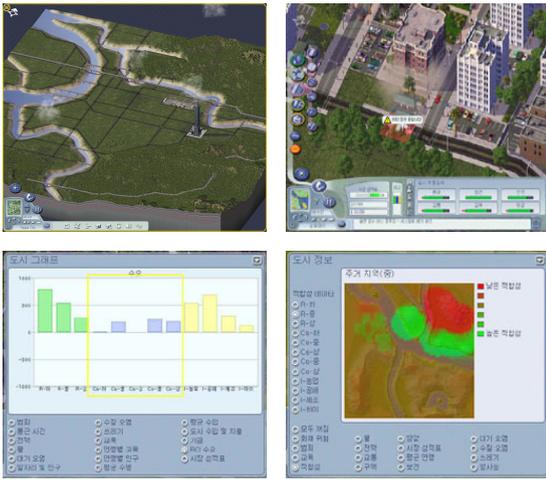
2.1 심시티(Simcity)

심시티[3]는 미국의 맥시스(Maxis)가 1989년에 개발한 도시개발 시물레이션 게임이다. 사용자는 시장이 되어 도시에 많은 시민을 유치하는 것이 목적이고, 그러한 과정에서 도시를 만드는데 필요한 요소들을 배치하면서 도시를 시물레이션 가능하다.

가장 기본적인 것은 생활구역 시물레이션이다. 주거, 상업, 공업 지대로 나눌 수 있는데 구역이 도시에서 얼마나 차지하냐에 따라서 직중에 따른 인구수라든지 환경오염등을 측정가능하다. 그리고 도로 배치에 따라서 도시의 교통흐름을 알 수 있고, 인구수가 많은 도심지역에서는 교통정체가 일어 날 수 있다. 이러한 도로는 현실의 도로와 마찰

가지로 유지비와 건설하는데 드는 비용이 있기 때문에 도시의 예산을 고려하여 배치해야한다. 그리고 그 외 수도관이나 전력선 혹은 경찰서나 소방서, 학교등과 같은 관공서 등을 배치할 수 있는데 이러한 시설들은 도시의 생활수준을 시뮬레이션 가능하게 한다.

이러한 심시티는 진행과정에서 도시가 변하는 모습을 살펴볼 수 있으며, 녹화와 차트기능이 지원이되어 도시를 개발하면서 생길 수 있는 문제점등을 파악하고 분석을 할 수 있다. 이러한 시뮬레이션 요소를 가지고 있어 심시티 후속작 중 하나인 심시티3000은 미국의 몇몇 고등학교에서 교재로 사용하기도 하였다. 국내에서는 이러한 심시티를 모방하여 베투얼 서울[4]이라는 서울건설 시뮬레이션을 개발 하기도 하였다.



(그림 1) 도시건설 시뮬레이터 심시티4

2.2 심즈(The Sims)

심즈[5]는 맥스에서 2000년에 발매된 생활 시뮬레이션 게임이다. 심즈는 심시티에서 사는 시민의 생활을 다루는 것인데 심시티와 다른게 도시가 아니라 도시 속에 들어 있는 거주 공간을 배경으로 삼고 있다. 그렇기 때문에 생활에 필요로하는 각종 생활용품과 집을 짓는 오브젝트들이 제공이 된다.

이러한 시뮬레이션은 다양한 오브젝트들이 준비가 되어 있어 가상으로 자신의 집을 인테리어 하는데 쓰이기도 한다.



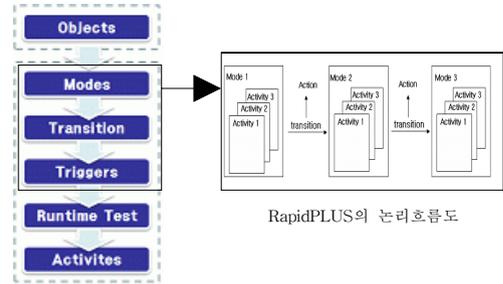
(그림 2) 생활 시뮬레이터 심즈2

3. 시스템 설계 및 구현

UI기반의 개발도구인 RapidPLUS를 이용하여 센서 네트워크의 센서 노드의 송수신상태와 데이터 전송률, 그리고 수신거리등의 정보를 가지는 오브젝트를 만들고, 위에서 소개한 S/W들처럼 센서 네트워크를 배치하여 보이지 않는 센서 정보들을 시뮬레이션을 통하여 볼 수 있도록 센서 노드를 프로토타이핑 한다.

3.1 RapidPLUS

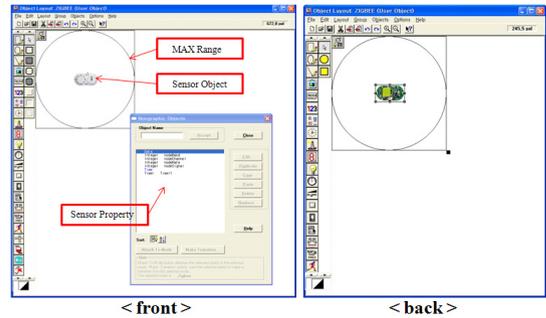
RapidPLUS[6]는 (그림 3)의 처리순서에 따라 시스템을 개발한다.



(그림 3) RapidPLUS의 개발 순서

3.2 센서 노드

Object Layout의 객체를 이용하여 (그림 4)와 같이 센서 노드의 기본 외형과 속성을 정의해준다.



(그림 4) 센서 노드

센서 노드는 Frequency Band, Channel, Data Rate등과 같은 속성을 설정해준다. 또한 Circle Object를 추가시켜 센서 노드의 최대 적용거리를 정의해준다.

이렇게 설정된 객체는 UDO(User Define Object)로 저장한다. 이렇게 UDO로 작성을 하면 RapidPLUS에서는 Active X나 JavaBean처럼 컴포넌트로 이용이 가능하다.

지그비 센서 노드는 Coordinator(ZC), Router(ZR), End Device(ZED)등의 3개의 Device Types를 가지고 있다. 이러한 Device들은 State Machine의 동작이 다르기 때문에 RapidPLUS의 Logic Editor로 각각의 동작을 정의를 해주고 이들을 서로 다른 UDO로 저장해준다. (그림 5)는 ZC는 보라색, ZR은 노란색, ZED는 녹색으로 구분하고 동작을 정의해준 모습이다.

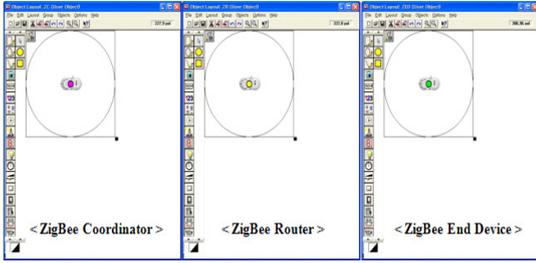
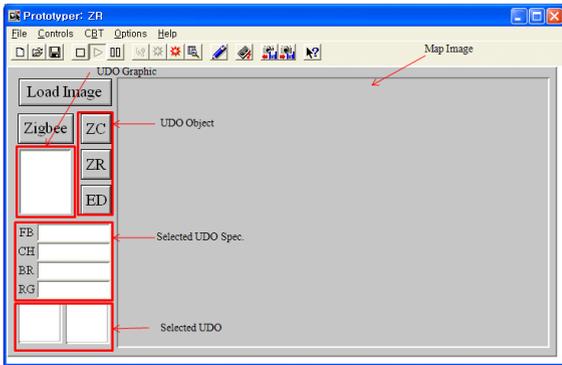


그림 5 지그비 Device UDO

3.3 시뮬레이션 인터페이스

Object Layout에서 기본으로 주어지는 Object들과 지금까지 구현한 UDO를 바탕으로 센서 노드가 동작 가능한 시뮬레이션 인터페이스를 (그림 6)과 같이 구현하였다.



(그림 6) 시뮬레이션 인터페이스

시뮬레이터의 기능은 다음과 같다. Load Image는 센서 네트워크를 구축할 대상이 되는 이미지를 불러오는 기능을 하고, UDO Object를 선택하면 UDO Graphic에 해당되는 UDO가 표시되고 UDO UDO Graphic에 표시된 UDO를 Drag & Drop을 이용하여 Map Image창으로 옮겨서 배치한다.

Map Image에서 UDO를 선택하면 UDO Spec과 UDO에 해당되는 UDO의 성능과 이름을 볼 수 있다. Spec창에서는 해당 UDO의 성능을 변경 가능하다.

4. 시뮬레이션 테스트

(그림 7)은 시뮬레이터의 동작과정을 보여준다. 처음 프로토타이퍼를 실행시키면 1번과 같은 화면이 보여지고 여기서 Load Image를 누르면 2번처럼 미리 지정된 Map Image가 화면에 보여진다. Zigbee를 누르고 ZC, ZR, ZED에 해당되는 객체를 선택하여 누르면 3번처럼 센서노드가 나타나고 그 센서 노드를 Drag & Drop을 이용해서 4번처럼 Map에 옮긴다. Map에 옮겨진 센서 노드에는 속성이 값이 존재하고 있지 않기 때문에 낮은 채도의 색상을 띄우고 있지만 속성이 FB, CH, BR, RG를 채워주면 5번처럼

활성화되며 수신범위를 원형으로 나타내는 것을 볼 수 있다.



(그림 7) 시뮬레이션 테스트

5. 결론

본 논문에서는 센서 네트워크의 센서 노드의 통신상태 기능을 RapidPLUS를 이용하여 구현하여보았다.

센서 네트워크를 구축함에 있어서 2장에서 언급한 S/W들을 통해서 주변 환경을 구축하고 3장에서 구현한 센서 노드들을 배치하여 시뮬레이션을 한다면 센서 노드의 적절한 위치와 필요한 개수를 파악할 수 있기 때문에 불필요한 인력낭비나 추가적인 센서 노드를 생산하는 문제점을 해결할 수 있을 것이다.

앞으로의 과제는 단순 평면에 배치하는 센서 노드에서 벗어나 3D작업을 통한 3D공간에 배치하고 그에 따른 통신상태를 시뮬레이션 하는 것과 지그비의 통신 상태뿐만 아니라 센서의 범위와 다른 객체로 표시할 수 있는 Image Editor기능등을 추가하는 것과 통신이나 센싱을 방해하는 장애요소인 객체들등을 추가하여 센서 네트워크를 구축함에 있어서 장애요소등을 예측하고 적절한 센서 노드의 위치선정이 가능하도록 하는 것이라 할 수 있고, 센서 노드가 동작하는 모양과 데이터에 있어서는 오차가 존재하기 때문에, NS-2등과 같이 검증된 네트워크 시뮬레이터등을 이용하여 피드백을 거쳐 점차적으로 개선을 시켜야 할 필요성이 있다.

참고문헌

- [1] Mark Weiser, "The Computer for the 21st Century," Scientific American Sep. 1991.
- [2] Mark Weiser, "Some Computer Science Problems in Ubiquitous Computing," Communications of the ACM, pp.75-84, July, 1993
- [3] <http://simcity.ea.com/>
- [4] <http://www.metro.seoul.kr/kor2000/seoulguide/newvirtual/index.cfm>
- [5] <http://thesims2.ea.com/>
- [6] <http://www.skymobilemedia.com/>