센서네트워크에서 노드의 소프트웨어 변경에 대한 가시성을 향상시키기 위한 버전관리기법*

정선우°, 김동규, 정기원
숭실대학교 대학원 컴퓨터학과
sunoo7906@paran.com°, topazz19@hanmail.net, chong@ssu.ac.kr

Version Management Method for Improving Visibility of Software Change of Nodes in Sensor Network

Sunwoo Jung°, Dongkyu Kim, Kiwon Chong
Department of Computing, Graduate School, Soongsil University

요 약

본 논문은 한국전자통신연구원에서 개발된 나노큐플러스 운영체제기반의 센서네트워크에서 각 노드의 응용모듈에 대한 버전을 관리하기 위한 기법을 제안한다. 제안한 기법을 응용모듈의 버전을 관리하기 위하여 고유식별번호를 각 노드의 헤더파일에 저장한다. 관리자 또는 개발자가 헤더파일에 저장되어 있는 각 노드의 고유한 식별번호를 사용하여 별도의 저장소에 저장되어 있는 노드의 형상정보를 한눈에 알아볼 수 있는 버전관리 기법을 제시하였다. 제안한 버전관리 기법을 나노큐플러스 운영체제기반의 센서네트워크 응용모듈에 적용하면 개발자 입장에서 각각의 응용모듈 변경에 대한 버전관리가 용이해지고, 형상항목인 소스코드간의 연관관계 및 변경된 소스코드의 버전에 대한 가시성을 향상시킬 수 있을 것으로 기대한다.

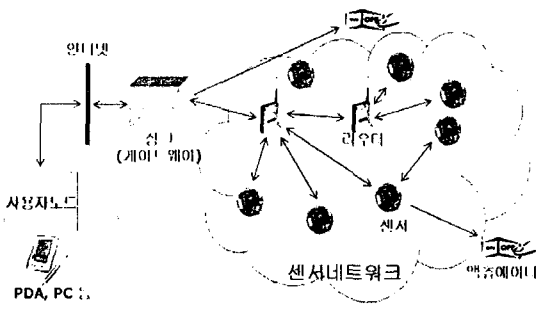
1. 서 론

센서네트워크는 센서가 부착되어 있어서 정보 수집이 가능하고 수집된 정보를 가공할 수 있는 프로세서가 장착되어 있으며 이를 전송할 수 있는 무선 송수신기를 갖춘 소형장치, 즉, 센서노드로 구성된 네트워크를 의미한다[1]. u-센서네트워크(Ubiquitous SensorNetwork)는 향후 IT분야의 핵심 인프라 산업으로써, 현재 다양한 연구가 활발하게 진행되고 있다. 센서네트워크는 여러 산업분야에 적용하기가 용이하며, 주기적으로 데이터를 수집하고, 수집된 데이터를 분석하여 시스템의 상황에 맞게 동작하는 자동화를 가능하게 하는 기술이다. 센서네트워크는 가정에서 원격검침이나 여러 사용자에 맞는 환경을 제공할 뿐만 아니라, 사람이 접근하기 어렵거나 위험한 지역의 데이터를 수집할 수 있는 장점이 있다. 이러한 센서네트워크의 여러 응용모듈들을 더욱 효율적으로 관리하기 위한 적절한 형상관리가 이루어지지 않을 경우, 응용모듈 구현 및 효과적인 센서네트워크 유지에 장애가 발생할 수 있다. 현재 한국전자통신연구원에서 개발된 센서네트워크 운영체제인 나노큐플러스를 기반으로 하는 응용모듈에 대한 형상관리에 대한 연구가 충분히 이루어지고 있지 않다. 따라서 본 논문에서는 노드의 응용모듈 변경에 따른 버전관리를 용이하게 할 수 있으며, 소스코드의 변경에 대한 가시성을 높일 수 있는 센서네트워크 응용모듈에 대한 버전관리기법을 제안한다.

2. 관련연구

2.1 센서네트워크

센서네트워크란 인간의 오감(시각, 청각, 촉각, 후각, 미각)을 대신하여 물리계 또는 환경계의 현상을 정량적으로 측정하여 정보를 검출하는 소자인 센서들이 통신기능을 지녀 네트워크를 이루는 것을 의미한다[2]. 센서네트워크 구조는 애드혹(Ad-hoc) 네트워크의 일종으로 센서를 이용하여 특정 목적을 위하여 [그림 1]과 같이 여러 센서노드들과 싱크노드의 크게 두 가지 노드로 구성되어 있다. 이 가운데 센서노드는 센서들을 사용하여 정보를 수집하고, 무선송수신기를 사용하여 통신을 한다. 그리고 싱크노드는 센서노드에서 보내지는 정보들을 사용자에게 전달해주는 중간자 역할을 수행하는 노드이다. 그리고 이들 중간에서 센서노드에서 최종 싱크노드까지의 가장 적절한 통신경로를 결정해주는 다리 역할을 하는 라우터노드가 있으며, 센서노드로부터 수집된 정보를 기반으로 싱크노드나 사용자의 통제를 통하여 원격검침이나 집안 조명의 밝기 조정 등의 실제 동작을 취하는 액츄에이터노드로 구성된다.



[그림 1] 센서네트워크의 구조

센서네트워크는 유비쿼터스 컴퓨팅에서 하나의 핵심기술로 작은 범위에서 동작하는 RF(Radio Frequency) 통신

* 본 연구는 숭실대학교 교내연구비 지원으로 이루어졌음

매체를 통해 연결된 무선 지능 센서들은 유비쿼터스 컴퓨팅 환경에서 사람과 컴퓨터 사이에 중개인으로써의 역할을 수행한다. 기존의 컴퓨팅 플랫폼과 달리 센서 노드들은 다음과 같은 세 가지 특징을 가진다[3].

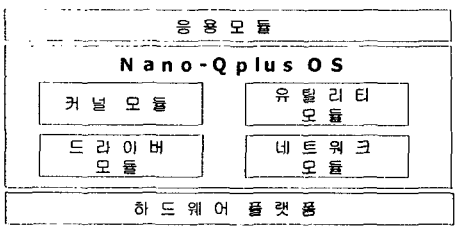
- 센싱정보에 대한 처리 능력과 저장메모리, 배터리 등의 모든 자원이 매우 제한적이다.
- 센서 네트워크를 플랫폼으로 보는 데이터 중심형(data-centric) 프로그래밍 스타일이 일반적이다.
- 재사용을 고려하지 않는 일회용 컴퓨팅 플랫폼이다.

센서 네트워크의 응용분야에는 산악, 강, 대기 등의 오염도를 측정, 군사 작전 시 적군 출현 관찰, 아주 큰 사파리공원에서 동물 관찰, 혹은 국립공원내의 산불 관찰 등이 있다. 센서 네트워크는 인간이 접근하기 힘든 지역의 인간의 눈 역할을 하는 작업을 수행하게 된다. 인간의 눈은 아니지만 센서를 통해 수집된 정보를 분석하여 자동으로 액츄에이터가 적당한 임무를 지능적으로 수행하도록 하면 보다 완벽한 센서네트워크를 구축하게 된다.

2.2. 나노큐플러스

나노큐플러스는 유아나 노인, 장애인을 위한 길안내 및 위험상황 정보 제공용으로 상용화할 수 있는 수준의 무선 센서 네트워크용 초소형 운용체제(OS)로 개발되었고, 한국전자통신연구소(ETRI)에서 정보통신부의 선도기반기술 개발사업의 지원을 받아 향후 유비쿼터스 환경 구축에 없어서는 안 될 10KB 미만의 초소형 OS인 '나노큐플러스 1.0' 을 2004년 4월초부터 작업을 시작하여 2005년 4월 24일에 개발되었으며 현재 1.6.1e버전에 이르고 있다. 나노큐플러스는 소형 센서 네트워크 시스템 개발을 위한 OS, 무선통신, 목표 시스템 설정 및 설치, 소스 편집, 모니터링 등의 기능을 수행하는 SW 기술로, 국방·디지털홈·의료·환경·건설 분야 등 산업 전반에 걸쳐 적용 가능하다. 여러 응용 분야에 적합한 초소형(10KB미만커널), 분산.실시간, 스마트(상황인식) 운영체제 기술을 갖추고 있으며, 유비쿼터스 센서 네트워크(USN)를 구축하기 위한 나노OS 플랫폼 제공한다. 나노큐플러스 OS 구조 및 형태는 다음과 같다[4].

- 센서 및 엑츄에이터의 종류에 따라 OS 커널을 최적화하여 재구성 가능한 Scalable OS(3~12KB미만의 커널 크기, Linux Kconfig script 사용 Rapid prototyping 지원)
- 다양한 스케줄러(FIFO, Preemption, 등) 지원 및 무선 통신 기능(RF, ZigBee,등) 지원
- 표준형 및 마이크로 임베디드 OS와 동일한 API subset 지원(POSIX 표준 기반)
- 유비쿼터스 센서 네트워크를 구축하기 위한 소형크기(4X6 cm)의 스마트 센서노드 기술

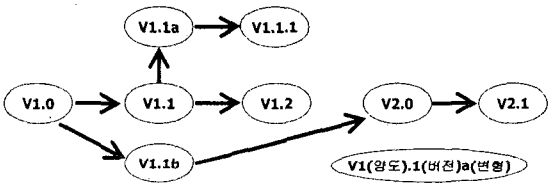


[그림 2] 나노큐플러스기반 응용모듈의 구성

2.3. 버전관리

버전관리에 대한 기존 방법으로 UNIX 버전관리 시스템인 RCS(Revision Control System)가 있다. 이는 오래전부터 사용된 도구이지만 현재에도 광범위하게 사용된다. 기준 버전에서부터 차이 값(델타)만 저장함으로써 디스크 요구사항을 최소화 시킨다. 이전 시스템 버전을 재작성하기 위해 델타 값을 최근 릴리스에 적용한다. 생성을 위해 어떻게 명명된 버전이나 릴리스도 허용한다. 서로 다른 릴리스의 독립적인 개발을 허용한다. 하지만 이 방법은 코드제어 시스템으로 설계되어 있어서 ASCII 문자들을 가지고 사용되며, 비 ASCII 표현으로 된 문서나 목적 코드를 관리하는데 이용될 수 없다. 또한 텍스트 기반의 사용자 인터페이스를 가지고 있으며, 버전 탐색의 어려움과 버전속성이 아니라 이름에 기본을 둔 버전 검색을 한다는 제약점을 가지고 있다[5,6].

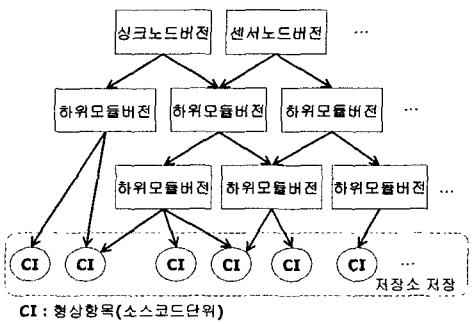
또한 [그림 3]과 같이 일반적으로 버전 넘버링은 크게 3가지로 매기게 된다. 개발팀에서 실제 사용자에게 배포되는 양도(release), 기존 시스템과 기능적으로 차이가 있는 버전(version), 시스템과 기능적으로는 차이가 없지만 처리속도, 성능 등과 같은 비기능적인 차이가 있는 변형(variant)으로 버전 넘버링을 한다[7]. 향후 연구에서도 이 같은 버전 넘버링을 이용하여 구현을 할 예정이다.



[그림 3] 버전넘버링 구조

3. 센서네트워크에서 노드의 버전관리

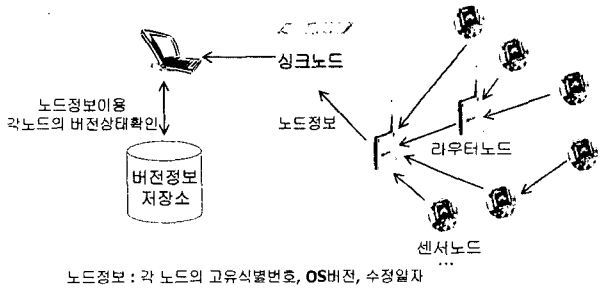
센서노드 내의 저장메모리의 크기가 제한적인 센서네트워크에서 개발자는 개발 중에 기능의 추가나 수정된 여러 노드의 변경에 대한 소프트웨어의 형상정보를 개발자의 저장소(repository)에 저장하고 있으며, 각 노드의 헤더파일에 고유식별번호를 저장하여 통신을 통해 사용자모드와 연결될 때 그 식별번호를 가지고 미리 저장해 둔 형상정보를 참조하여 각 노드의 버전상태와 업데이트 유무에 대한 개발자의 가시성을 향상시킬 수 있는 방법을 제시한다.



[그림 4] 계층적인 구조의 형상항목

여러 노드의 응용모듈의 소프트웨어 버전정보는 각 노드의 하위모듈과 그 모듈의 형상항목으로 계층적으로 버전을 관리하고, 그러기 위해서는 노드의 형상항목(소스코드)의 변경에 따른 버전을 관리를 해야 하는데 [그림 4]에서 보는 바와 같이 각 노드는 여러 하위모듈들로 구성되어 있으며 다시 그 하위모듈에는 여러 형상항목으로 이루어져 있다. 여기서 말하는 형상항목에는 소스코드, 문서, 테스트 집합 등을 의미하지만, 본 논문에서 개발자 입장에서 구현에 대한 형상항목을 소스코드 단위로만을 고려하였으며 소스코드의 변경이 노드의 기능 향상이나 새로운 기능을 추가함으로써 개발자의 의사에 의해 버전정보를 넘버링을 하여 저장소에 저장한다[8].

이외에도 응용모듈의 소프트웨어 변경에 대한 관리를 위하여 [그림 5]와 같이 버전정보 저장소에 저장되는 속성에는 노드명, 노드의 고유식별번호, 변경일자, 형상항목의 주요기능에 대한 설명, 형상항목의 변경전·현재의 버전, 해당항목이 참고하는 항목, 변경 전과 기능상의 추가된 사항과 변경된 기능에 대한 설명 등이 있다.



[그림 5] 버전정보 저장소를 사용한 응용모듈의 버전관리

각 노드의 버전관리 헤더파일에 [그림 6]과 같이 노드의 고유식별번호와 OS버전, 수정일자를 저장하여 노드와 사용자 모드의 버전정보저장소에 저장된 각 노드의 버전정보를 얻어 그 정보를 화면에 보여줌으로써 개발자의 각 노드의 형상정보 및 버전 관리의 가시성을 향상시킬 수 있고 이후 노드의 기능 확장성 및 형상항목의 버전관리가 용이해진다. 즉 변경관리에서 변경상태를 추적하기 위해 각각의 노드정보를 이용하여 개발자의 버전정보저장소에 저장되어 있는 버전상태를 자동적으로 변경 정보를 화면에 나타내어 줌으로써 개발자가 한눈에 변경상태를 알 수 있으며 노드의 버전상태를 체크하여 항상 최신의 버전을 유지하는데 도움을 준다.

```
////////////////////////////////////////
/*
 * Version management headers:
 */
#define Node_Identifier SENSOR_11
#define NanoOS_Version  qplusn-1.6.1e
#define Modified_Date   2005/09/05
////////////////////////////////////////
```

[그림 6] 노드 응용모듈의 헤더파일에 삽입한 버전정보

4. 결론 및 향후연구

본 논문에서 센서네트워크를 구성하는 노드들의 응용모듈의 버전관리에 대한 소프트웨어 변경에 대한 가시성을 향상시키는 버전관리 방법을 제안하였다. 각 노드에 응용모듈의 버전정보 및 변경정보를 저장하기에는 노드의 메모리가 제한적이기 때문에 각 노드 형상항목을 관리하기 위하여 개발자 환경에 노드의 형상관리 저장소를 만들어 각 노드의 헤더파일에 정의된 노드의 고유식별번호를 이용하여 저장소에 저장된 변경 정보를 알 수 있어서 각 노드의 상태정보를 알 수 있다. 실제 나노큐플러스 기반의 센서네트워크 구조에서는 이와 같은 버전관리가 되어 있지 않아 본 논문의 방법을 적용하면 개발자의 응용모듈변경에 대한 관리가 용이해지고, 변경관리에 대한 가시성을 향상시킬 수 있다.

향후 연구에서는 실제로 본 논문의 아이디어를 바탕으로 소프트웨어 변경이 일어난 노드의 변경정보를 관리하기 위한 형상관리 툴 개발할 예정이다. 또한 보다 안정화된 노드의 OS환경이 구축된다면 위에서 제안한 실제노드의 고유 정보인 노드의 고유식별번호를 이용하여 해당 노드의 버전정보를 알아내고 그 노드의 버전에 대한 업데이트가 필요한 노드에 자동 업데이트할 수 있는 에이전트 프로그램에 관한 연구를 수행할 계획이다. 이러한 자동 업데이트는 각 노드마다의 선택 모드 설정할 수 있다. 예를 들어 항상 최근 버전의 프로그램이 필요한 노드가 있을 수 있으며, 일정한 주기별로 업데이트가 필요한 노드, 마지막으로 반드시 사용자에 의해 업데이트 요청에 대한 수동적인 노드마다의 설정에 따라 버전 관리를 자동화할 수 있을 것이다.

5. 참고문헌

[1] 채동현, 한규호, 임경수, 안순신, "센서네트워크의 개요 및 기술동향," 정보과학회지 제22권 제12호, pp.5-12, 2004
[2] 표철식, u-센서네트워크, ETRI, 2004.10.29
[3]http://redwood.snu.ac.kr/PAPERS/source/domestic/03-kiss-fall-SenOS.pdf
[4] http://se.ssu.ac.kr/kylee/
[5] http://powerdb.net/study/cvs/rcs-1.html
[6]http://doc.kldp.org/KoreanDoc/html/CVS_Tutorial-KLDP/index.html
[7] Ivan Sommerville, "Software Engineering 5th," Addison-Wesley, 1996
[8] Hagen Volzer, Anthony MacDonald, Brenton Atchison, Andrew Hanlon, Peter Lindsay, Paul Strooper, "SubCM: A Tool for Improved Visibility of Software Change in an Industrial Setting," IEEE Computer Society, Oct,2004