

유비쿼터스 센서 네트워크를 위한 디렉토리 서비스 설계 및 구현

박홍재* , 김영만* , 한재일* , 이용준** , 오세원**
*국민대학교 컴퓨터공학부, **한국전자통신연구원

The Design and Implementation of the Ubiquitous Sensor Network Directory Service

Park, Hongjae* , Kim Youngman* , Han Jaeil* , Lee Yongjoon* , Oh Sewon*
* Kookmin University, * Electronics and Telecommunications Research Institute
E-mail : hjpark0@kookmin.ac.kr, ymkim@kookmin.ac.kr

요 약

최근 유비쿼터스 환경에서 센서 네트워크를 이용한 여러 가지 연구가 활발히 진행되고 있다. 향후 USN 인프라 구축이 완료됨에 따라 USN을 활용하기 위한 USN 어플리케이션 개발을 보다 손쉽게 하기 위해 USN 미들웨어(USN Middleware)의 필요성이 대두되며, 디렉토리 서비스(Directory Service)는 USN 미들웨어를 구성하는 핵심 서비스 중의 하나이다. USN 인프라가 인터넷과 결합된 USN 환경에서의 디렉토리 서비스는 USN 계층의 자원에 대한 개별적인 속성을 돌려 줄 수 있어야 한다. 그러나 풍부한 컴퓨팅 자원, 높은 대역폭(bandwidth) 및 짧은 지연시간, 멀티캐스트 기능 등을 가지고 있는 인터넷과 달리 USN을 구성요소인 WSN(Wireless Sensor Network)은 제한된 컴퓨팅 자원, 에너지 문제, 낮은 대역폭, 무선 통신의 불안정성 등의 특성을 가지고 있다. 이러한 이유로 디렉토리 서비스 기술은 WSN에 직접 적용하기가 어렵다. 본 논문에서는 기존의 디렉토리 서비스에서 나타날 수 있는 문제와 WSN의 특성을 고려하여 USN 인프라에 적합한 USN 디렉토리 서비스(Ubiquitous sensor network Directory Service; UDS)를 설계 및 구현한다.

1. 서 론

언제 어디서든 인간의 편의를 도모하고자 하는 유비쿼터스 센서 네트워크(USN) 시대에 발맞춰 많은 연구들이 이루어지고 있다. 향후 USN 인프라 구축이 완료됨에 따라 USN을 활용하기 위한 USN 어플리케이션 개발을 보다 손쉽게 하기 위해 USN 미들웨어(USN Middleware)의 필요성이 대두되며 [1], 디렉토리 서비스(Directory Service)는 USN 미들웨어를 구성하는 핵심 서비스 중의 하나이다.

USN 인프라가 인터넷과 결합된 USN 환경에서의 디렉토리 서비스는 USN 계층의 자원에 대한 개별적인 속성을 돌려 줄 수 있어야 한다. 그러나 풍부한 컴퓨팅 자원, 높은 대역폭(bandwidth) 및 짧은 지연시간, 멀티캐스트 기능 등을 가지고 있는 인터넷과 달리 USN을 구성요소인 WSN(Wireless Sensor Network)은 제한된 컴퓨팅 자원, 에너지 문제, 낮은 대역폭, 무선 통신의 불안정성 등의 특성을 가지고 있다. 이러한 이유로 디렉토리 서

비스 기술은 WSN 에 직접 적용하기가 어렵다. 본 논문에서는 기존의 디렉토리 서비스에서 나타날 수 있는 문제와 WSN 의 특성을 고려하여 USN 인프라에 적합한 USN 디렉토리 서비스(Ubiquitous sensor network Directory Service; UDS)를 설계 및 구현한다.

본 논문의 구성은 다음과 같다. 2 장에서는 디렉토리 서비스와 관련한 국내외 연구 사례를 살펴보고, 3 장에서는 UDS 설계 그리고 4 장에서는 UDS 구현에 대하여 기술하며, 마지막으로 5 장에서는 결론 및 향후 과제에 대해 논한다.

2. 관련연구

네이밍 서비스는 통상적으로 알려져 있는 개념으로써 클라이언트가 어떤 자원의 이름을 주었을 때 그 자원에 대한 속성(예, 주소, 파일 경로 등)을 돌려주는 서비스이다. 이와 유사하게 인식되고 있는 디렉토리 서비스는 클라이언트가 하나 이상의 어떤 속성을 주었을 때 이 속성을 가지고 있는 각 자원의 여러 속성을 돌려주는 서비스로서 속성 기반 네이밍 서비스(attribute-based naming service)로 부르기도 하며 보통 네이밍 서비스를 포함한다[2][3][4][5].

디렉토리 서비스를 위해서는 USN 계층에 존재하는 다양한 USN 정보자원들을 식별하기 위한 체계가 필요하다. 이러한 식별체계를 위하여 국내외에서 여러 연구 및 표준화가 진행 중이다. 먼저 RFID 영역에서는 물류 정보를 표현하기 위해 XML 을 기반으로 하여 설계된 Physical Markup Language(PML)[6]이 있다. 다음으로 센서 네트워크 영역에서는 스마트 센서의 인터페이스를 위하여 국제 표준단체인 IEEE 에서 진행중인 IEEE 1451[7]이 있으며, 센서에서 발생하는 센싱 데이터뿐만 아니라 센서 자체에 관한 정보를 표현하기 위하여 XML 에 기반하여 설계된 SensorML[8]이 있다. 또한 국내에서는 한국정보사회진흥원을 중심으로 한 센서 관련 정보의 표준화를 USIS

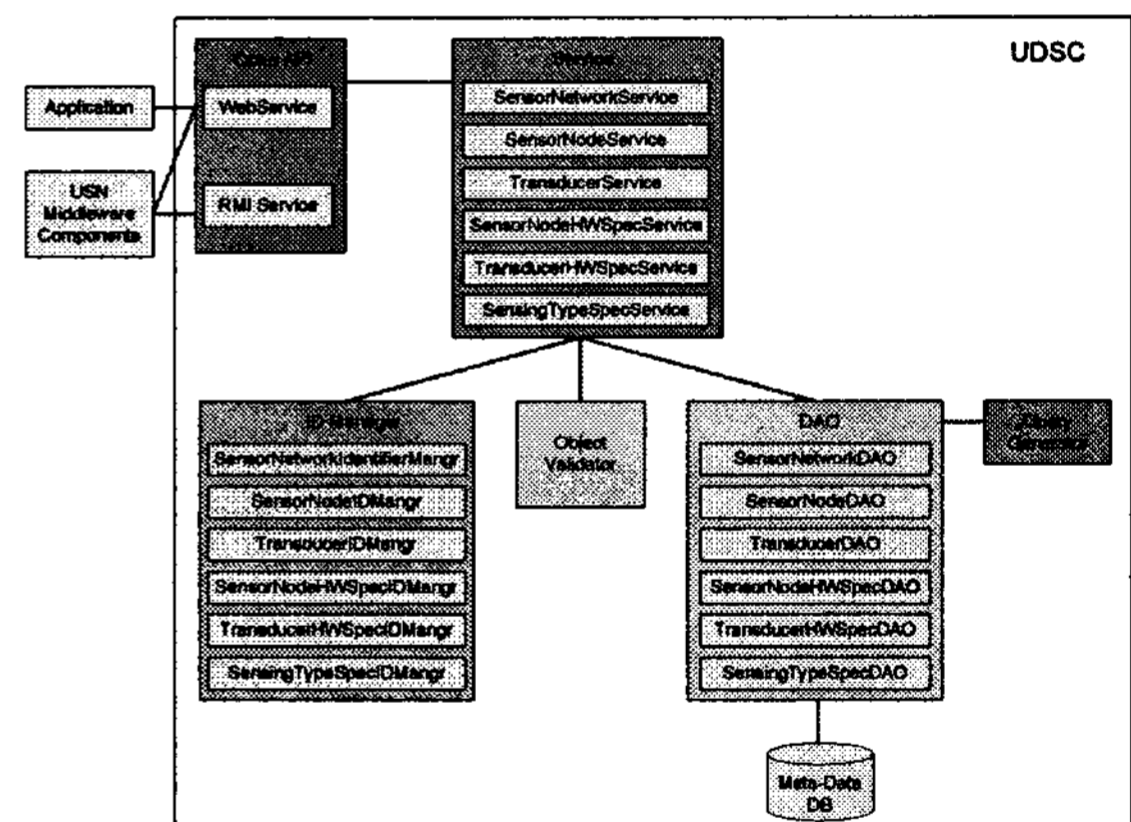
(Ubiquitous Sensor Information System)[9]라는 이름으로 진행 중이다.

이와 같이 USN 환경에서의 디렉토리 서비스에 기본적으로 필요한 USN 계층의 개별 자원 식별 및 표준화 방안에 대한 연구가 국내·외적으로 진행되고 있으나, USN 인프라와 인터넷이 결합된 USN 환경에서 클라이언트에게 USN 계층의 개별 자원에 대한 속성을 돌려 줄 수 있는 디렉토리 서비스 연구는 국내·외에서 아직 보이지 않고 있다.

3. UDS(USN Directory Service) 설계

UDS 는 UDS 외부의 사용자에게 Webservice 와 RMI 서비스를 제공한다. 본 장에서는 UDS 의 구조와 외부 요청 처리절차 및 응답처리 절차를 설계한다.

3.1 UDS 구조



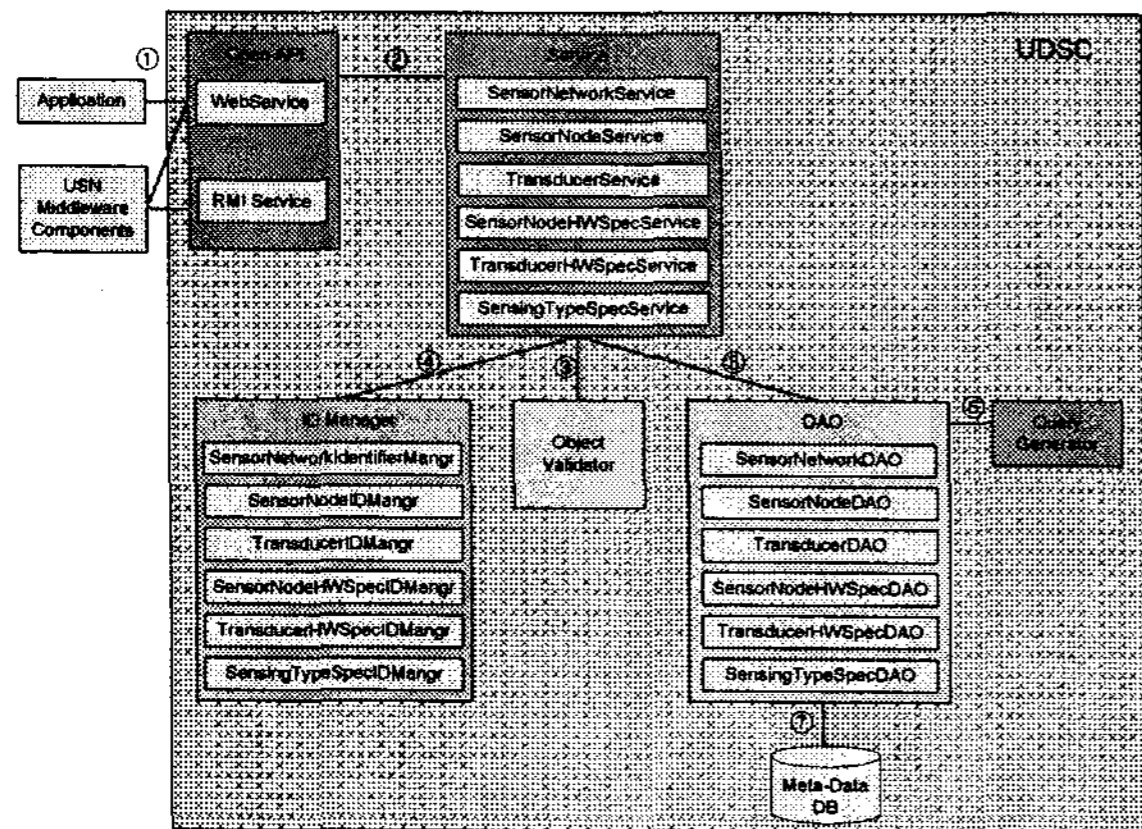
(그림 1) UDS 구조

UDS 는 USN 미들웨어와 연결된 무선 센서 네트워크에 관한 USN 자원을 등록/ 삭제/ 갱신/ 수정/ 조회하는 기능을 제공하는 시스템으로서 Open API, Service, ID Manager, Object Validator, DAO, Query Generator 로 구성된다. 각 구성요소별 역할은 다음과 같다.

USN 자원의 등록/ 삭제/ 갱신/ 수정/ 조회기능을 사용하는 UDS 외부의 사용자들과의 인터페이스는 Open API 가 담당한다. Open API 의 각 함수와 연관된 DAO 를 호출하는 역할은 Service 가 담당한다.

다. USN 자원을 데이터베이스에 저장/ 조회/ 삭제 하는 기능은 DAO 가 담당한다. 객체의 유효성 검증은 Object Validator 가 담당한다. DAO 의 속성 기반 조회관련 요청시 요청과 매핑되는 질의어를 생성하는 역할은 Query Generator 가 담당한다. 마지막으로 USN 자원 식별자 발급 기능은 ID Manager 가 담당한다.

3.2 요청처리 절차

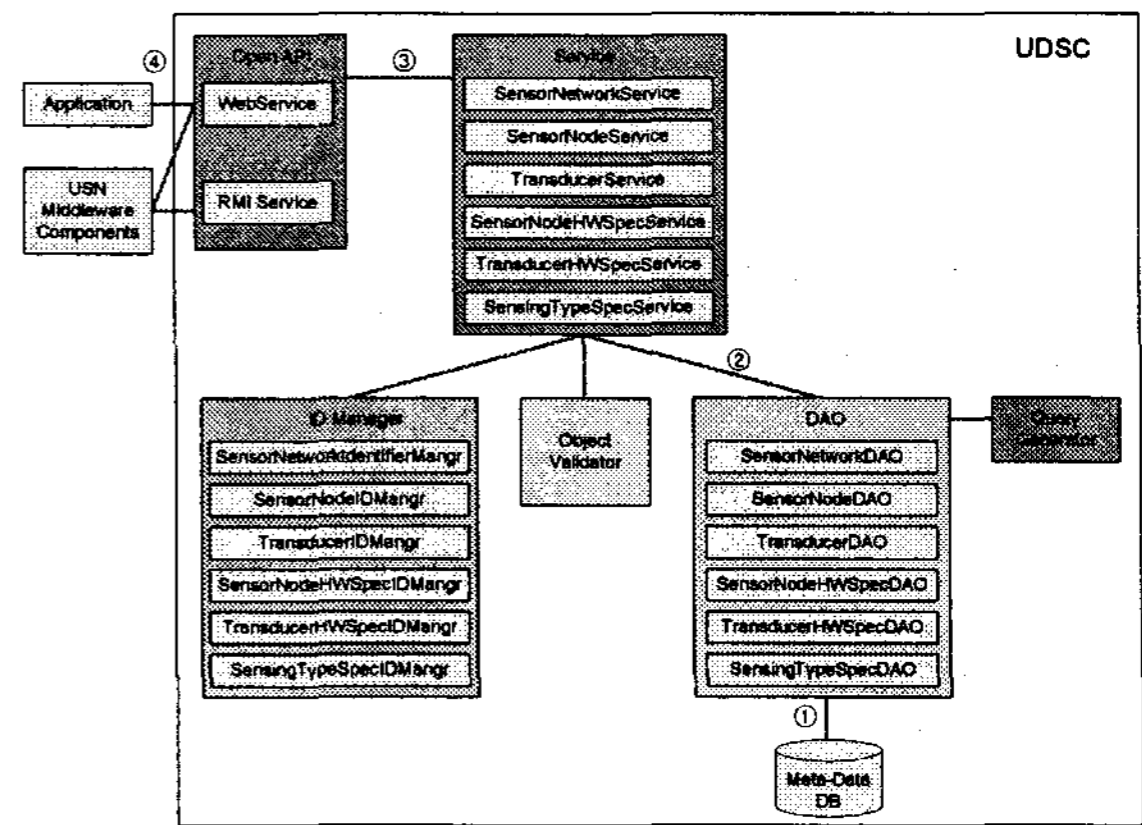


(그림 2) UDS 요청처리 절차

외부 사용자의 요청은 Open API 인 WebService 또는 RMI Service 의 호출로 시작된다. 요청 호출을 받은 Open API 는 6 가지 Service 중에 하나로 전달되며, Service 는 파라미터로 전달되는 객체의 유효성을 검사하기 위하여 Object Validator 를 호출한다. 객체의 유효성이 입증되면 USN 자원 식별자를 생성 또는 검사하기 위하여 ID Manager 를 호출하며, 위 과정이 에러없이 수행되었다면 6 가지 DAO 중 하나로 전달된다. DAO 에서는 SQL 문을 생성하기 위하여 Query Generator 를 호출하며 최종적으로 Meta-Data DB 에 변경 또는 조회 등의 질의문을 수행한다.

3.3 응답처리 절차

외부 사용자 요청에 대한 응답 처리는 Meta-Data DB 에서 수행한 SQL 질의문의 결과를 DAO 가 수집하여 Service 로 전달하며 Open API 를 통해 최종적으로 사용자에게 전달된다.



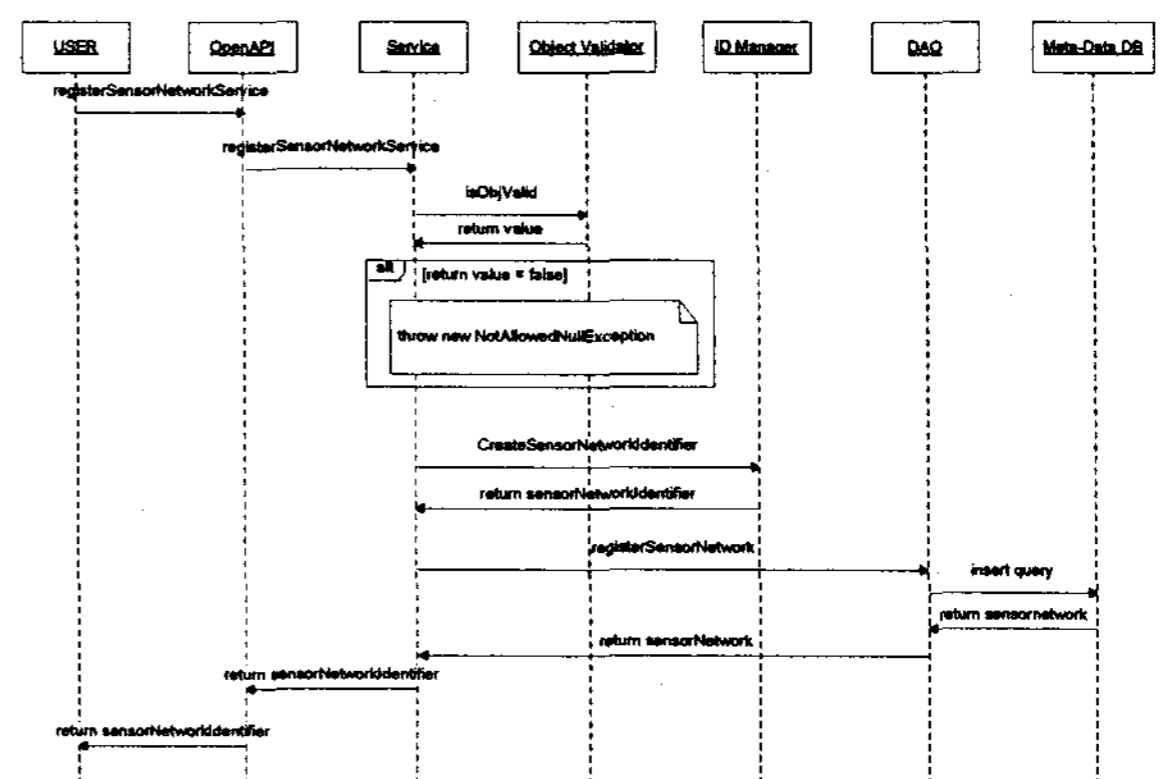
(그림 3) UDS 응답처리 절차

4. UDS 구현

UDS 는 JDK 1.5 [10]기반에서 동작하며, 비즈니스 로직은 스프링 프레임워크(Spring Framework) [11]를 사용하며, OR 매핑으로는 하이버네이트 프레임워크(Hibernate Framework)[12]를 사용한다. 그리고 메타데이터를 저장하는 데이터 베이스는 MySQL 5.0[13]을 사용하였으며, 자카르타 아파치 톰캣(Jakarta Apache Tomcat) [14]기반에서 구동한다.

UDS 의 동작과정은 등록, 삭제, 갱신, 수정, 조회로 나뉘며 각 과정별 시퀀스 다이어그램은 다음과 같다.

4.1 등록



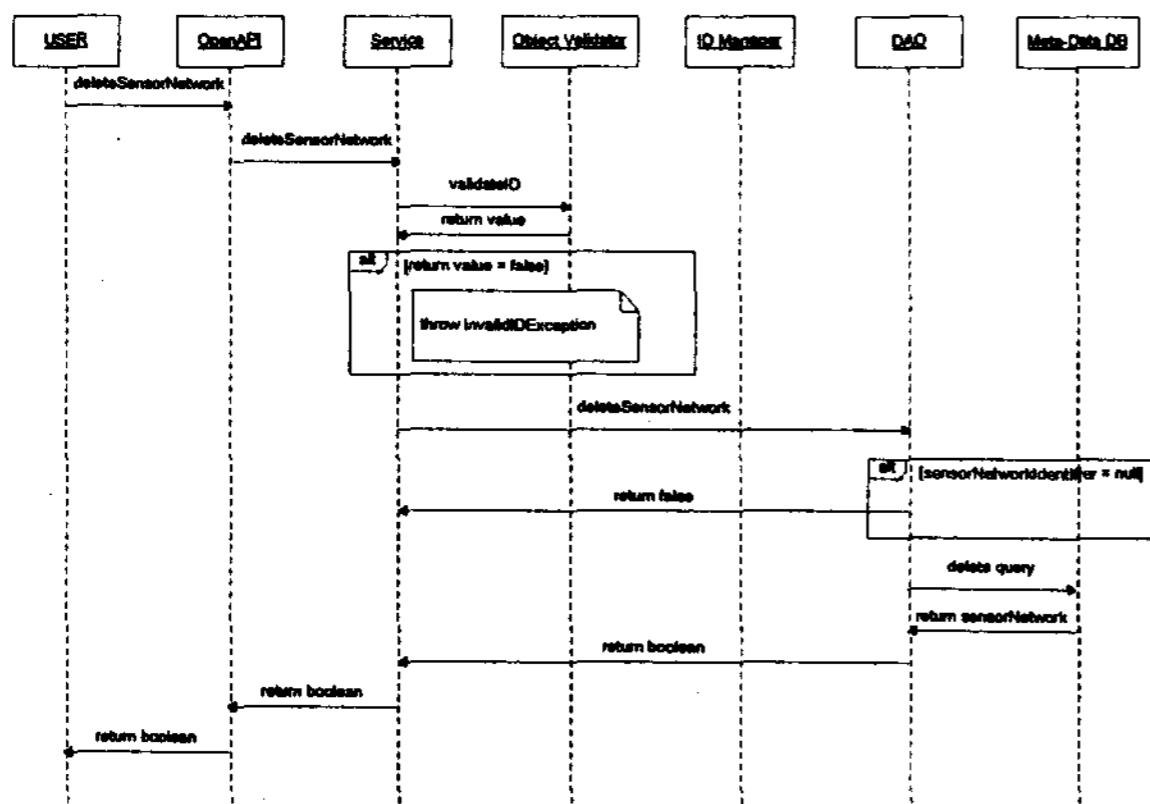
(그림 4) 등록 시퀀스 다이어그램

USN 자원의 등록을 원하는 사용자는 UDS 의 open API 를 호출하여 등록을 요청한다. UDSC

open API 내부에서는 Service 컴포넌트의 API 를 호출하며, Service 컴포넌트 내부에서는 유효성검사 컴포넌트(Object Validator)를 통해 등록할 USN 자원 객체의 유효성을 검사하고 이상이 없을 시 ID Manager 컴포넌트를 통해 등록할 USN 자원의 식별자를 발급받는다. UDS open API 는 DAO 컴포넌트의 API 를 호출해 실제 데이터베이스에 USN 자원의 등록을 요청하고 등록이 완료되면 등록된 USN 자원의 식별자를 반환한다((그림 4) 참조).

4.2 삭제

USN 자원의 삭제를 원하는 사용자는 UDS 의 open API 를 호출하여 삭제를 요청한다. UDS open API 에서는 Service 컴포넌트의 API 를 호출하고, Service 컴포넌트 내부에서는 유효성검사 컴포넌트를 통해 삭제할 USN 자원 식별자의 유효성을 검사하고 DAO 컴포넌트의 API 를 호출해 데이터베이스에서 USN 자원을 삭제한다((그림 5) 참조).

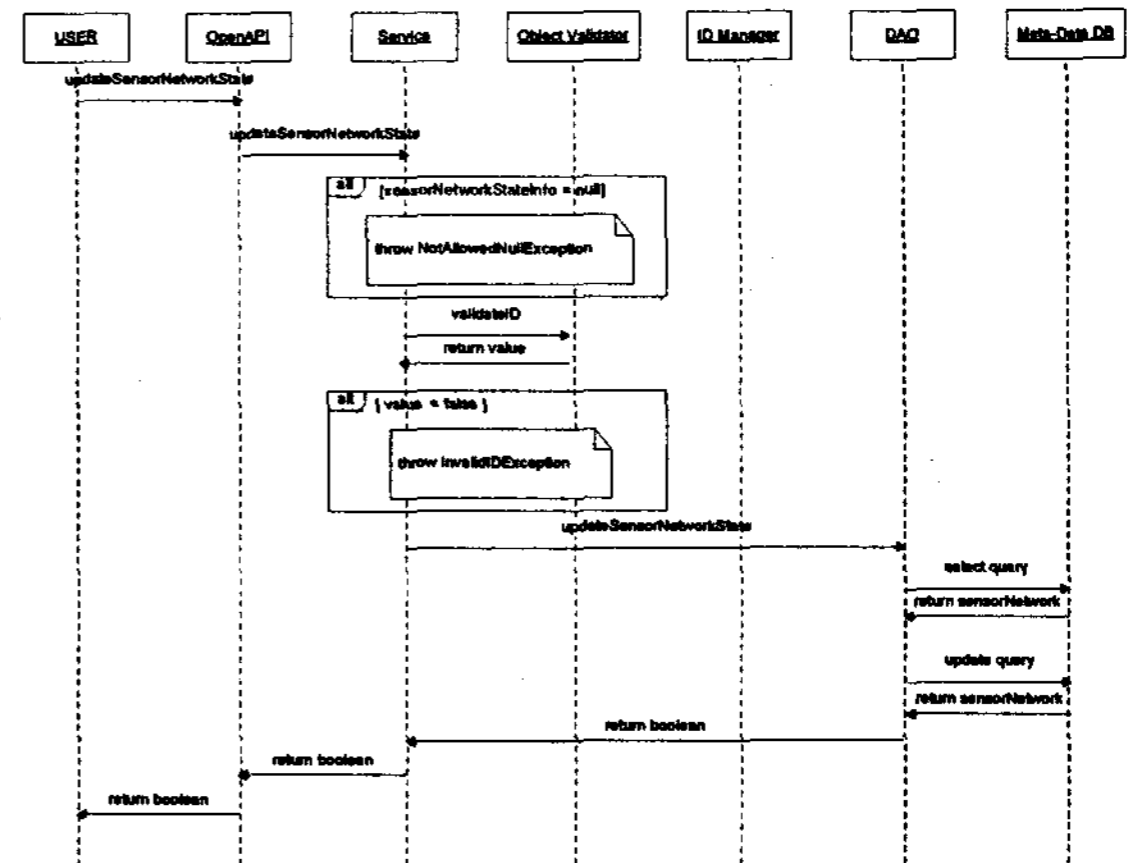


(그림 5) 삭제 시퀀스 다이어그램

4.3 갱신

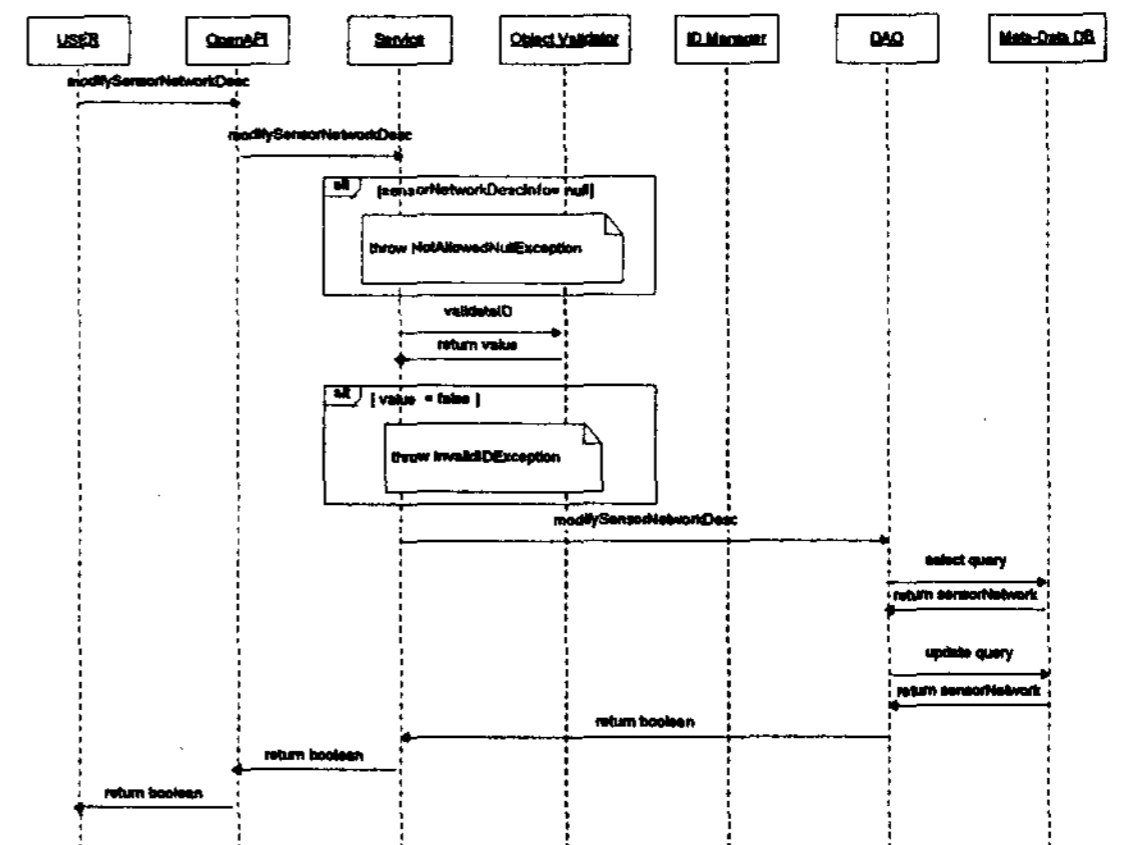
USN 자원의 갱신을 원하는 사용자는 UDS 의 open API 를 호출하여 갱신을 요청한다. UDS open API 에서는 Service 컴포넌트의 API 를 호출하고, Service 컴포넌트 내부에서는 유효성검사 컴포넌트를 통해 갱신할 USN 자원 식별자의 유효성을 검사하고 이상이 없을 시 DAO 컴포넌트의 API 를 호

출해 데이터베이스의 USN 자원이 갱신되도록 요청한다. USN 자원 갱신은 센서네트워크의 경우 센서 네트워크에 포함된 센서노드의 수, 센서노드의 경우 센서노드에 포함된 트랜스듀서의 수 등의 동적인 정보들에 대하여 수행한다((그림 6) 참조).



(그림 6) 갱신 시퀀스 다이어그램

4.4 수정

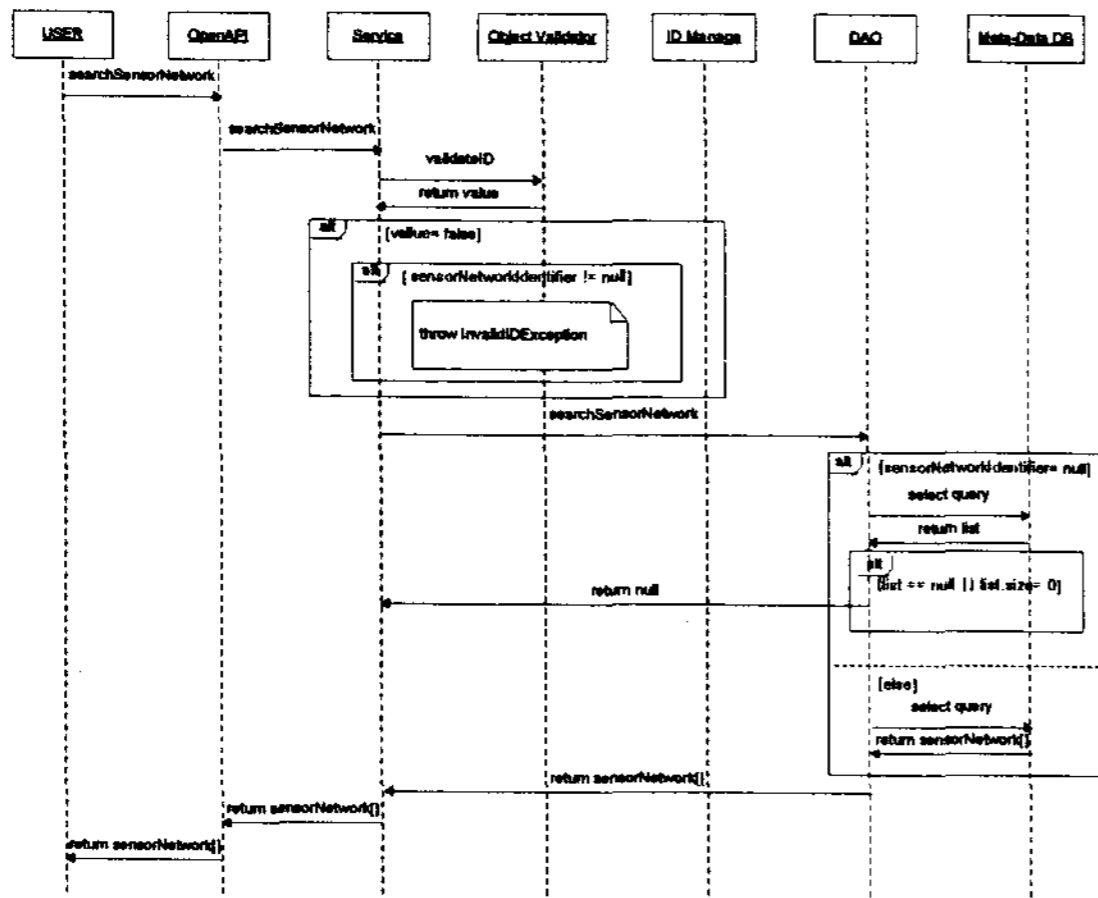


(그림 7) 수정 시퀀스 다이어그램

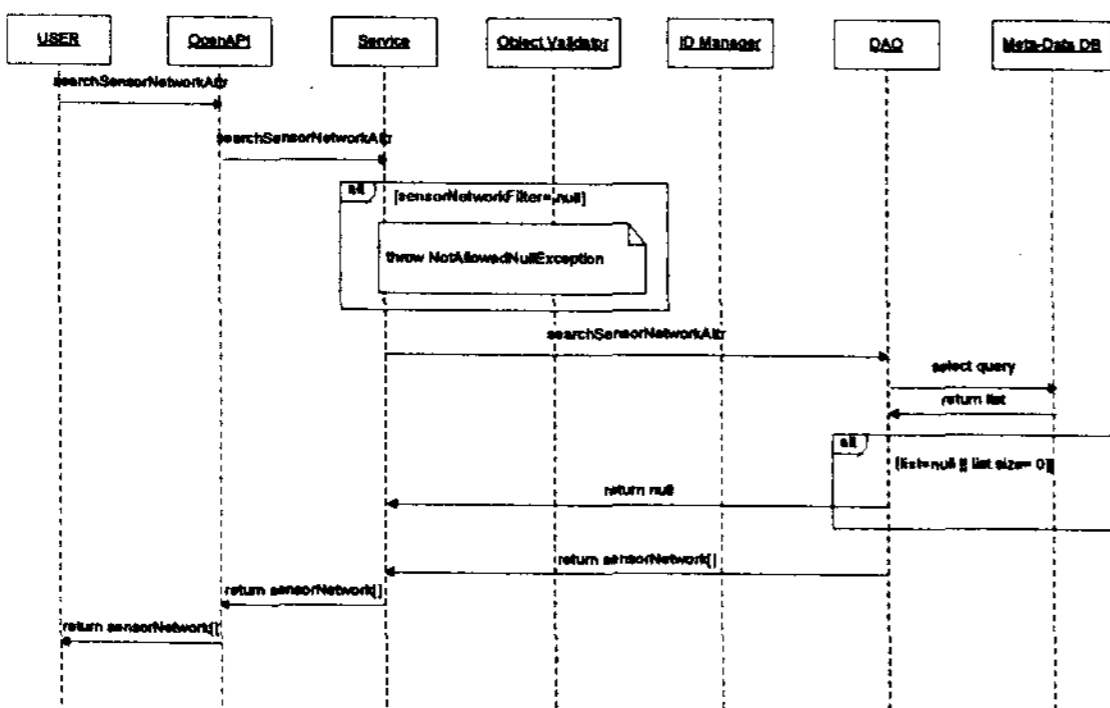
USN 자원의 수정을 원하는 관리자나 USN 미들웨어 컴포넌트는 UDS 의 open API 를 호출하여 수정을 요청한다. UDS open API 에서는 Service 컴포넌트의 API 를 호출하고, Service 컴포넌트 내부에서는 유효성검사 컴포넌트를 통해 수정할 USN 자원 식별자의 유효성을 검사하고 이상이 없을 시 DAO 컴포넌트의 API 를 호출해 데이터베이스의 USN 자원 정보가 수정되도록 요청한다. 수정 가능한

USN 자원의 정보로는 센서네트워크, 센서노드 그리고 트랜스듀서의 이름, 기능 등의 정적인 정보들이다 ((그림 7) 참조).

4.5 조회



(그림 8) ID 기반 조회 시퀀스 다이어그램



(그림 9) 속성기반 조회 시퀀스 다이어그램

USN 자원의 조회를 원하는 조회자나 USN 미들웨어 컴포넌트는 UDS의 open API를 호출하여 조회를 요청한다. UDS open API에서는 Service 컴포넌트의 API를 호출하고, Service 컴포넌트 내부에서는 DAO 컴포넌트의 API를 호출해 실제로 데이터베이스에서 USN 자원이 조회되도록 요청하고 조회된 USN 자원 리스트를 조회자나 USN 미들웨어 컴포넌트에게 반환한다. USN 자원 조회에는 두가지 형태의 조회, ID 기반 조회와 속성 기반 조회가 있다((그림 8, 9) 참조).

4.6 메타데이터

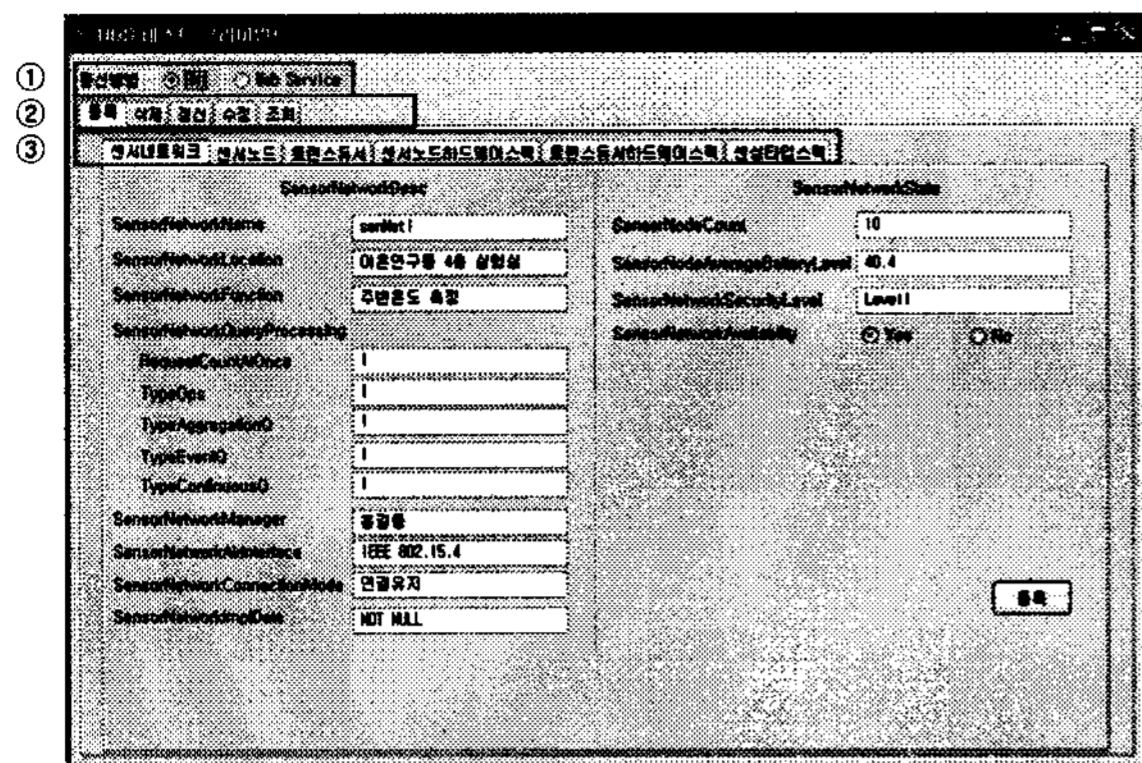
USN 자원을 표현하는 메타데이터는 [표 1]과 같다.

<표 1> USN 메타데이터

USN 자원	메타데이터
sensorNetwork	센서네트워크ID, 센서네트워크 이름, 구축위치, 주요기능 설명, 동시 처리 가능한 질의 개수, 질의처리 Operator 수행 가능 여부, Aggregation/ Event /Continuous 질의 처리 수행 가능 여부, 관리자 정보, 구축시점, 센서네트워크 접속 가능 여부, 통신프로토콜 방식, 접속 모드, 센서노드 개수, 평균 전원잔량, 보안 레벨, 센서네트워크 모니터링 주기, 센서네트워크 모니터링 응답 제한시간
sensorNode	센서노드ID, 주요기능, 위치, SW 버전, 동작유무(0:off, 1:on), 전원잔량, 트랜스듀서 개수
sensorNodeHWSpec	센서노드 HWSpec ID, 센서노드 하드웨어 스펙 프로파일 번호, 제조사, 모델명, 전원공급방식
transducer	트랜스듀서ID, 타입(0:sensor, 1:actuator), 트랜스듀서 이름, 동작유무, 센서 상태, 센싱타입ID, 액추에이터 동작단계
transducerHWSpec	트랜스듀서HWSpec ID, 타입(0:센서, 1:액추에이터), 트랜스듀서 하드웨어스펙 프로파일 번호, 트랜스듀서 제조사, 트랜스듀서 모델명, 센서기능, 센싱값 범위, 오차허용 범위, 센서모드, 센싱 단위, 제어대상, 동작 단위
sensingTypeSpec	센싱타입ID, 센싱타입 스펙의 프로파일 번호, 센싱타입 이름, 센싱값의 단위, 센싱 데이터 종류, 센싱 데이터 크기, 센싱주기

4.7 실행 결과

UDS를 시험하기 위하여 SWT 기반의 테스트 프로그램을 작성하였으며, 이 프로그램의 구성은 크게 ①통신방법, ②Open API 패널, ③USN 자원 패널로 나뉜다. 먼저 상단의 통신방법은 UDS가 제공하는 두가지 방식의 Open API 서비스인 WebService와 RMI를 선택할 수 있는 라디오 버튼으로 구성된다. 다음으로 Open API의 기능인 등록, 삭제, 갱신, 수정, 조회를 선택할 수 있는 탭으로 구성되며, 마지막으로 UDS 자원의 메타데이터로 구성된 탭이 존재한다.



[그림 10] 테스트 프로그램 실행결과 화면

5. 결론 및 향후과제

급속히 성장하고 있는 USN 시장에서 USN 자원에 대한 하드웨어 및 운영체제에 대한 기술은 외국 기업 및 연구단체들이 국내에 비해 매우 앞서 있는 것이 사실이다. 하지만 국외에서도 아직 USN 미들웨어에 대한 상용화된 기술 및 모델이 존재하지 않으며, 관련 연구는 시작단계이므로 국내외 업체 및 연구단체들의 경쟁이 치열할 것으로 예상된다. 그러므로 본 논문에서 설계 및 구현한 USN 미들웨어의 핵심 기술 중에 하나인 USN 디렉토리 서비스를 활용하여 USN 미들웨어에 대한 국제적인 경쟁력을 가지고 USN 표준화 및 응용 서비스의 상용화 작업에 커다란 영향을 미칠 수 있을 것이다.

향후 USN 미들웨어의 구성요소들을 설계 및 구현한 후 테스트 베드를 구축하여 본 논문에서 제안한 USN 디렉토리 서비스의 유효성을 입증할 예정이다.

[참고문헌]

- [1] 김영만, "USN 인프라 서비스 지원을 위한 계층적 미들웨어 플랫폼 구조", 한국정보처리학회지, vol.12 no.5 pp.52-65. 2005
- [2] Birrell, A., Nelson, G. and Owicki, S. (1993). Network objects. In proceedings 14th ACM Symposium on Operating Systems Principles, pp.217-30.

[3] www.microsoft.com I, Microsoft Corporation. Active Directory Services.

[4] ISO/IEC 9594-1, "Information technology - Open Systems Interconnection - The Directory: Overview of concepts, models and services", Fifth edition, 2005-12-15

[5] ISO/IEC 9594-2, "Information technology - Open Systems Interconnection - The Directory: Models", Fifth edition, 2005-12-15

[6] PML Core Specification 1.0, Auto-ID Center Recommendation 15 September 2003

[7] IEEE 1451 and related websites:
<http://ieeel451.nist.gov/>

[8] SensorML and related websites:
<http://vast.uah.edu/SensorML/>

[9] 이재호, 안철현, 김유정, 신상철, 이재용, "국가적 USN 정보자원 관리체계 구축 정책에 관한 연구" 심사중, 한국전산원 정보화정책, 2006.

[10] <http://java.sun.com/javase/index.jsp>

[11] <http://www.springframework.org>

[12] <http://www.hibernate.org>

[13] <http://www.mysql.com/>

[14] <http://tomcat.apache.org/>