

모바일 센서 노드 지원을 위한 USN 디렉토리 서비스

주수현*, 박민수*, 오대성*, 김민기*, 한재일*

*국민대학교 컴퓨터 공학과

USN Directory Service For Mobile Sensor Node

Ju Su-Hyun*, Park Min-Su, Oh Dae-Sung, Kim Min-Ki, Han Jae-Il

Dept. of Computer Science, Kookmin University

E-mail : emtype@paran.com

요 약

다가올 유비쿼터스 시대를 선도할 기술 중 하나인 USN(Ubiquitous Sensor Network)은 산업 전반에 커다란 변화를 가져올 것으로 기대된다. USN은 각종 센서 노드로부터 사물 및 환경 정보를 감지, 저장, 가공, 통합하고 상황인식 정보 및 지식 콘텐츠 생성을 통하여 언제, 어디서, 누구나 원하는 맞춤형 서비스를 자유로이 이용할 수 있는 첨단 지능형사회의 기반 인프라이다. USN과 더불어 이를 활용하기 위한 USN 어플리케이션의 개발들이 늘고 있으며 이에 USN 미들웨어의 필요성 또한 증가하고 있다. USN 디렉토리 서비스는 USN 미들웨어를 구성하는 핵심 서비스 중 하나이며, 지역적으로 분산된 USN 자원에 대한 정보를 관리하고 USN 자원에 대한 검색을 실시간으로 제공하는 서비스이다. USN 자원은 고정된 자원과 이동성을 지닌 자원으로 나눌 수 있다. 하지만 현재 이동성을 지닌 센서 노드(Mobile Sensor Node)를 지원할 수 있는 USN 디렉토리 서비스는 보이지 않고 있다.

본 논문은 이러한 Mobile Sensor Node의 특성을 지원하는 USN 디렉토리 서비스의 모델들을 제시하고 그에 대한 설계 및 구현을 논한다.

I. 서론

다가올 유비쿼터스 시대를 선도할 기술 중 하나인 USN(Ubiquitous Sensor Network)은 산업 전반에 커다란 변화를 가져올 것으로 기대된다. USN은 각종 센서 노드로부터 사물 및 환경 정보를 감지, 저장, 가공, 통합하고 상황인식 정보 및 지식 콘텐츠 생성을 통하여 언제, 어디서, 누구나 원하는 맞춤형 서비스를 자유로이 이용할 수 있는 첨단 지능형사회의 기반 인프라이다.[1] USN과 더불어 이를 활용하기 위한 USN 어플리케이션의 개발들이 늘고 있고 USN 미들웨어의 필요성 또한 증가하고 있다. USN 디렉토리 서비스는 USN 디렉토리 서비스는 USN 미들웨어를 구성하는 핵심 서비스 중 하나이며, 지역적으로 분산된 USN 자원에 대한 정보를 관리하고 USN 자원에 대한 검색을 실시간으로 제공하는 서비스이다. 하지만 현재 이동성을 지닌 모바일 센서노드(Mobile Sensor Node)를 지원할 수 있는 USN 디렉토

리 서비스는 보이지 않고 있다. 본 논문은 이러한 모바일 센서 노드의 특성을 지원하는 USN 디렉토리 서비스의 모델들을 제시하고 그에 대한 설계 및 구현을 논한다.

본 논문의 구성은 다음과 같다. 2장에서는 관련연구 및 문제점에 대해 논하고 3장에서는 모바일 센서 노드를 지원하기 위해 제공해야 하는 서비스에 대해 논한다. 4장에서는 디렉토리 서비스 설계 및 구현에 대해 기술한다. 마지막으로 5장에서는 결론 및 향후과제에 대해 논한다.

II. 관련연구 및 문제점

다양한 분야의 USN 응용 서비스를 구축하기 위한 시스템의 개발은 USN 미들웨어 시스템의 발전에도 큰 영향을 끼치게 되었다. 실제로 USN 응용 서비스에 따라서 다양한 형태의 USN 미들웨어들이 소개되었는데, USN

응용 서비스의 QoS 요구조건에 대한 보장을 최우선 목표로 하여 개발된 MILAN 미들웨어[2], RFID 미들웨어와 유사하게 끊임없이 획득되는 센싱정보에 대하여 이벤트들을 설정함으로써 USN 응용 시스템에게 원하는 정보를 전송하는 이벤트 기반의 DSWare 미들웨어[3], USN 응용 서비스 변화 및 센서 네트워크 주변 환경 변화에 따라 센서노드 미들웨어의 기능을 무선 통신을 통하여 동적으로 변화시킬 수 있는 Impala 미들웨어[4], 센서 네트워크에 존재하는 센싱정보들을 분산 데이터베이스의 분산 데이터로 간주하여 USN 응용 시스템의 요구사항을 분산질의 처리 과정으로 수행할 수 있는 TinyDB[5] 등이 개발되어 왔다. 그러나 기존 USN 미들웨어의 디렉토리 서비스 기술은 동적인 USN 자원에 대한 지원을 하지 않고 있으며 이에 따라 동적인 USN 자원을 추적, 검색할 수 있는 식별체계와 메타데이터 정보를 검색, 수정, 갱신 삭제할 수 있는 USN 디렉토리 서비스 기술에 대한 심도 있는 연구가 필요하다.

III. 모바일 센서 노드를 지원하기 위해 요구되는 서비스

USN 미들웨어는 센서 네트워크 및 센서노드에 관한 메타정보를 효율적으로 유지하고, USN 응용 서비스 시스템에게 전달할 수 있어야 한다. USN 응용 서비스 시스템은 이러한 메타정보를 이용함으로써 다수의 센서 네트워크들이 복잡하게 동시에 연결되어 있는 USN 미들웨어로부터 자신이 원하는 정보만을 손쉽게 추출하여 획득할 수 있게 된다. USN 환경에서 이러한 메타정보로는 시간 흐름에 따라 변화가 없는 정적인 메타정보와 시간 흐름에 따라 변화가 발생하는 동적인 메타 정보로 나눌 수 있다.[2] 정적 메타 정보의 경우 USN 미들웨어의 관리도구를 통하여 관리될 수 있으나 동적 메타 정보의 경우는 정확한 정보를 유지하기 위해 USN 미들웨어가 센서 네트워크에 대하여 주기적인 모니터링 메시지를 전송하여 메타 정보를 효율적으로 획득하기 위한 방법을 반드시 제공해야한다.

디렉토리 서비스가 지원해야 하는 기본적인 서비스는 [표-1]과 같다.

| | | |
|--------|-----------------|---------------------|
| 등록 서비스 | 메타데이터를 등록하는 서비스 | |
| 조회 서비스 | ID기반 조회 서비스 | 메타데이터를 ID로 조회하는 서비스 |

| | | |
|--------|-----------------|----------------------|
| | 속성기반 조회 서비스 | 메타데이터를 속성으로 조회하는 서비스 |
| 삭제 서비스 | 메타데이터를 삭제하는 서비스 | |
| 수정 서비스 | 수정 서비스 | 정적인 메타데이터를 수정하는 서비스 |
| | 갱신 서비스 | 동적인 메타데이터를 갱신하는 서비스 |

[표-1] 기본적으로 제공하는 서비스

조회 서비스의 경우 ID로 조회를 할 것인지 속성으로 조회를 할 것인지에 따라 ID 기반 조회 서비스와 속성기반 조회 서비스로 나누어진다.

수정 서비스의 경우에 정적인 정보를 수정하는 수정서비스와 동적인 정보를 수정하는 갱신 서비스로 분류할 수 있다.

기본적인 디렉토리 서비스는 정적 메타데이터만 지원하기 때문에 동적 메타데이터 즉, 모바일 센서 노드를 지원할 수 있는 디렉토리 서비스는 [표-2]와 같은 서비스를 제공해야 한다.

| | |
|--------|-----------------|
| 등록 서비스 | 고정 센서노드 등록 서비스 |
| | 이동 센서노드 등록 서비스 |
| 조회 서비스 | 고정 센서 노드 조회 서비스 |
| | 이동 센서 노드 조회 서비스 |
| 삭제 서비스 | 고정 센서 노드 삭제 서비스 |
| | 이동 센서 노드 삭제 서비스 |
| 수정 서비스 | 고정 센서 노드 수정 서비스 |
| | 이동 센서 노드 수정 서비스 |
| 갱신 서비스 | 고정 센서 노드 갱신 서비스 |
| | 이동 센서 노드 갱신 서비스 |

[표-2] 모바일 센서 노드를 위해 제공하는 서비스

IV. 디렉토리 서비스 설계 및 구현

1. 설계 요구사항

모바일 센서 노드의 위치를 판별하기 위해선 USN 자원을 구별할 수 있는 식별자가 있어야 한다. 또한 식별자에 기반을 둔 메타데이터 모델이 있어야 하며, 이들 자원에 대한 등록, 삭제, 조회, 수정, 삭제 서비스를 제공해야한다.

본 논문은 IP Address에 기반을 둔 식별자를 정의한다.

또한 식별자에 기반을 둔 메타데이터 모델을 제시하며, 자원에 대한 등록, 삭제, 조회, 수정, 삭제 서비스를 제공한다.

2. 식별체계

자원의 식별체계는 [표-3]과 같이 총 128bit로 구성된다.

| Header | Address | Time Stamp | Sequence |
|--------|---------|------------|----------|
| 4 bit | 32 bit | 60 bit | 32 bit |

[표-3] 식별체계 구성

식별체계 중 첫 번째 4bit에 해당하는 Header 필드는 USN 자원을 식별하는 정보이다. 예를 들어 센서 네트워크일 경우 '0001', 센서 노드일 경우 '0010'의 값을 갖는다.

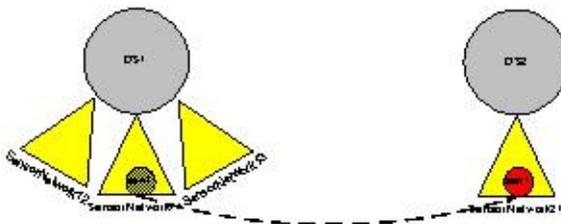
그 다음 32bit에 해당하는 Address 필드는 자원이 최초 등록된 디렉토리 서버의 IP Address 값을 갖는다.

다음 60bit인 Time Stamp 필드는 자원이 등록된 시간 정보이고, 마지막 32bit Sequence 필드는 등록된 시간이 같을 때 구별하기 위한 발급순서 정보이다. 위의 식별체계를 이용해 총 128bit의 코드를 16진수로 하여 32자리 아이디 발급할 경우 최초 등록지가 '192.168.0.4'인 센서 노드의 식별자는 '2c0a80040000112964e088000002d' 값을 가진다.

3. 디렉토리 서비스 모델

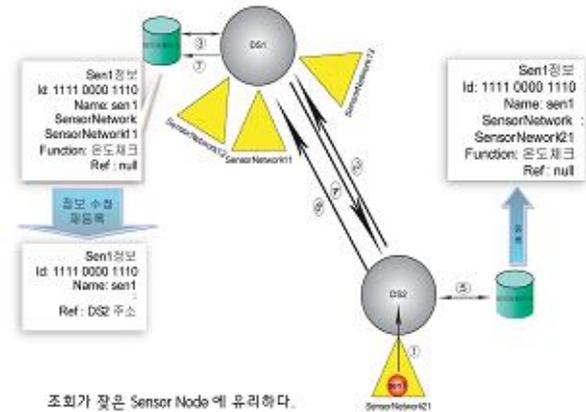
모바일 센서 노드가 다른 디렉토리 서버가 관리하는 센서 네트워크로 이동했을 경우 해당 센서 노드의 메타데이터를 옮겨간 디렉토리 서버로 복사할 것인지 아니면 복사를 하지 않고 reference 정보를 이용해 옮겨간 쪽 DB에서 원본 DB의 내용을 가리킬 것인지에 따라 모델 구성이 달라질 수 있다. 전자를 'DB Copy & Reference' 후자를 'Reference' 라 정의한다.

두 모델은 다음과 같이 'DS1'(디렉토리 서비스 이름)가 관리하는 센서 네트워크에 속해 있는 'sen1'이 'DS2'가 관리하는 센서 네트워크로 이동한 상황을 가정한다.



[그림-1] 센서 노드의 이동 가정

가. DB Copy & Reference



조회가 잦은 Sensor Node 에 유리하다.

[그림-2] DB Copy & Reference 모델

센서 노드가 가지는 메타데이터는 [표-4]와 같다.

| | |
|---------------|-------------------------|
| ID | 센서 노드의 식별자 |
| Name | 센서 노드의 이름 |
| SensorNetwork | 센서 노드가 속해 있는 센서 네트워크 이름 |
| Function | 센서 노드의 기능 |
| Ref | Reference 정보 |

[표-4] 센서 노드의 메타데이터

- ① DS2는 자신의 센서 네트워크에 센서 노드 sen1이 온 것을 감지하고 sen1의 ID를 받는다.
- ② DS2는 sen1의 ID를 분석해서 처음 등록된 곳이 DS1임을 알고 DS1에게 sen1의 메타데이터를 요청한다.
- ③ DS1은 sen1의 메타데이터와 그와 관련된 하드웨어 정보, 트랜스듀서 정보들을 자신의 데이터베이스에서 얻는다.
- ④ DS1은 데이터베이스에서 얻어온 메타데이터들을 DS2에 넘겨준다.
- ⑤ DS1로부터 받은 메타데이터들을 DS2가 자신의 데이터베이스에 넣는다.
- ⑥ 등록이 성공적으로 됐다면 DS1에 해당 센서 노드에 대한 수정 요청을 한다.
- ⑦ DS1은 자신이 가지고 있던 sen1와 관련된 트랜스듀서 정보를 모두 지우고 sen1의 메타데이터에는 ID와 reference(DS2의 주소)만을 남겨둔다.

① ~ ⑦의 과정을 거치고 나면 [표-5]와 같이 메타데이터가 변한다.

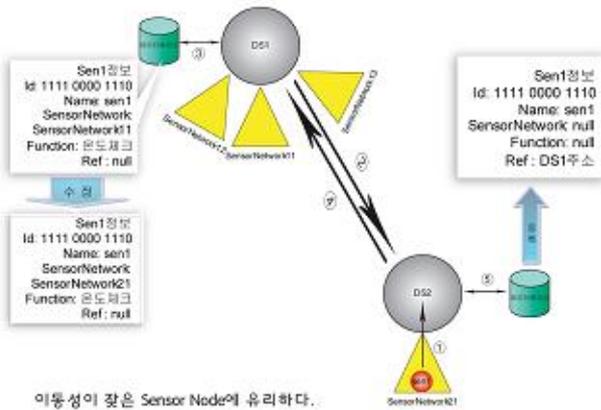
| | 이동 전 | | 이동 후 | |
|----------------|-----------------|-----|-----------------|-----------------|
| | DS1 | DS2 | DS1 | DS2 |
| ID | 0010... | 없음 | 0010... | 0010... |
| Name | sen1 | | null | sen1 |
| Sensor Network | sensor Network1 | | sensor Network2 | sensor Network2 |
| Function | 온도체크 | | null | 온도체크 |
| Ref | null | | DS2의 주소 | null |

[표-5] DB Copy & Reference 방식의 센서 노드 이동 후의 메타데이터 변화

| | 이동 전 | | 이동 후 | |
|----------------|-----------------|-----|-----------------|-----------------|
| | DS1 | DS2 | DS1 | DS2 |
| ID | 0010... | 없음 | 0010... | 0010... |
| Name | sen1 | | null | null |
| Sensor Network | sensor Network1 | | sensor Network2 | sensor Network2 |
| Function | 온도체크 | | 온도체크 | null |
| Ref | null | | null | DS1의 주소 |

[표-6] Reference 방식의 센서 노드 이동 후의 메타데이터 변화

나. Reference



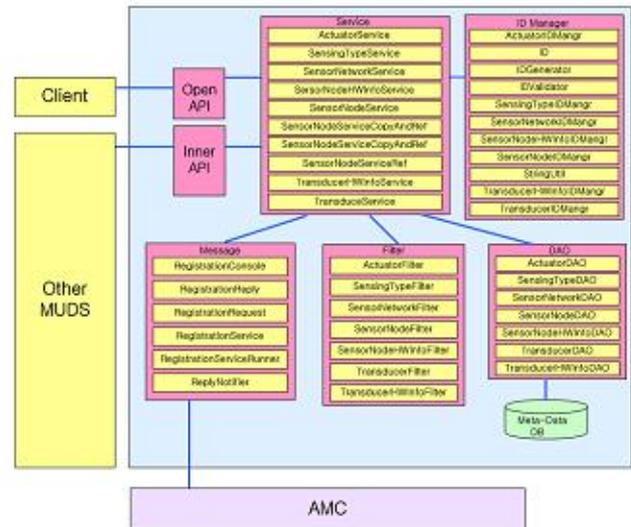
이동성이 잦은 Sensor Node에 유리하다.

[그림-3] Reference 모델

- ① Sensor Node 'sen1'은 'DS2'에 자신의 ID를 전달함으로써 이동해왔다는 것을 알린다.
- ② 'DS2'는 'sen1'의 ID를 분석해서 처음 등록된 곳이 'DS1'임을 알고 'DS1'에게 'sen1'의 메타데이터의 정보 중 SensorNetwork 정보를 'SensorNetwork2'로 수정할 것을 요청한다.
- ③ 'DS1'이 데이터베이스에서 'sen1'의 정보를 SensorNetwork 정보를 수정한다.
- ④ 확인 메시지를 DS2에 보낸다.
- ⑤ 'DS2'가 자신의 데이터베이스에 'sen1'의 정보를 reference가 DS1을 가리키게 해서 등록한다.

① ~ ⑤의 과정을 거치고 나면 [표-6]와 같이 메타데이터가 변한다.

4. 디렉토리 서비스의 구현



[그림-4] 디렉토리 서비스 시스템 구조

디렉토리 서비스의 시스템 구조는 [그림-8]와 같다.

가. Service

디렉토리 서비스가 클라이언트에게 제공하는 서비스이며 OpenAPI를 통해 사용할 수 있다.

① SensorNetworkService

센서 네트워크에 대한 요청을 처리하며 등록, 삭제, 수정, 갱신, 조회를 제공한다.

② SensorNodeService

센서 노드에 대한 요청을 처리하며 등록, 삭제, 수정, 갱신, 조회를 제공한다.

③ TransducerService

트랜스듀서에 대한 요청을 처리하며 등록, 삭제, 수정, 갱신, 조회를 제공한다.

④ SensingTypeService

센싱타입에 대한 요청을 처리하며 등록, 삭제, 수정, 조회를 제공한다.

⑤ ActuatorService

액추에이터에 대한 요청을 처리하며 등록, 삭제, 수정, 조회를 제공한다.

⑥ SensorNodeHWInfoService

센서 노드 하드웨어에 대한 요청을 처리하며 등록, 삭제, 수정, 조회를 제공한다.

⑦ TransducerHWInfoService

트랜스듀서 하드웨어에 대한 요청을 처리하며 등록, 삭제, 수정, 조회를 제공한다.

나. DAO

메타데이터 데이터베이스에 메타데이터 정보의 저장, 인출, 삭제 등의 작업을 제공한다.

① SensorNetworkDAO

메타데이터 데이터베이스에 센서 네트워크의 메타데이터 정보 처리 요청을 수행한다.

② SensorNodeDAO

메타데이터 데이터베이스에 센서 노드의 메타데이터 정보 처리 요청을 수행한다.

③ TransducerDAO

메타데이터 데이터베이스에 트랜스듀서의 메타데이터 정보 처리 요청을 수행한다.

④ SensingTypeDAO

메타데이터 데이터베이스에 센싱타입의 메타데이터 정보 처리 요청을 수행한다.

⑤ ActuatorDAO

메타데이터 데이터베이스에 액추에이터의 메타데이터 정보 처리 요청을 수행한다.

⑥ SensorNodeHWInfoDAO

메타데이터 데이터베이스에 센서노드 하드웨어의 메타데이터 정보 처리 요청을 수행한다.

⑦ TransducerHWInfoDAO

메타데이터 데이터베이스에 트랜스듀서 하드웨어의 메타데이터 정보 처리 요청을 수행한다.

다. ID Manager

센서 네트워크, 센서 노드 등의 ID를 생성한다.

라. Filter

속성기반 조회 시 사용하는 Filter를 제공한다.

마. Message

센서 노드가 이동해 왔을 때의 메타데이터 처리를 담당한다.

바. Open API

센서 네트워크, 센서 노드, 트랜스듀서 등의 등록, 조회, 삭제, 수정, 갱신을 위해 사용자에게 제공하는 API

4. 서비스 시나리오

두 모델의 제공하는 등록, 삭제, 조회, 수정, 갱신 서비스 중 지면관계상 센서 노드에 대한 조회 서비스 제공 과정을 살펴보도록 하겠다.

가. 정적 메타데이터 조회

클라이언트는 OpenAPI를 호출하여 조회 서비스를 요청한다. 디렉토리 서비스의 SensorNodeService는 해당 요청을 받아 ID의 Validation을 체크하고 이상이 없을 때에는 SensorNodeDAO를 호출하여 실제 메타데이터 정보를 얻어 온다. 그 후 얻어 온 메타데이터 정보를 클라이언트에게 돌려준다.

나. 동적 메타데이터 조회

클라이언트는 OpenAPI를 호출하여 조회 서비스를 요청한다. 디렉토리 서비스의 SensorNodeService는 해당 요청을 받아 ID의 Validation을 체크하고 이상이 없을 때에는 SensorNodeDAO를 호출하여 실제 메타데이터 정보를 얻어 온다. 얻어 온 메타데이터 정보에 Reference 정보가 있을 때에는 현재 디렉토리 서버에 해당 자원이 없는 것을 의미하므로 Reference에 있는 IP주소에 해당하는 디렉토리 서버의 InnerAPI를 호출하여 메타데이터 정보를 얻어온 후 그 정보를 클라이언트에게 돌려준다.

