

# 유비쿼터스 센서 네트워크를 위한 USN 메타데이터 정의 및 메타데이터 관리 시스템

박종현\*, 강지훈\*\*

## USN Metadata Definition and Metadata Management System for Ubiquitous Sensor Network

Jong-Hyun Park \*, Ji-Hoon Kang \*\*

### 요 약

유비쿼터스 센서 네트워크(USN) 환경은 다양한 센서들을 기반으로 사용자에게 고급의 서비스를 제공하는 것을 그 목적으로 한다. 이러한 환경에서 센서 및 센서 노드, 그리고 센서 네트워크는 이종으로 존재하며 그 특성 또한 매우 다양하다. 그러므로 이들 사이에 상호운용성을 위하여 단일의 메타데이터를 정의하고 이를 효과적으로 관리하는 것은 매우 중요한 일이다. 이를 위하여 OGC(Open Geospatial Consortium)에서는 센서를 모델링하기 위한 표준 언어로 SensorML(Sensor Model Language)을 제안하고 있다. 그러나 SensorML은 센서들 사이의 처리 모델을 기술하고, 이 때 필요한 각 센서들의 정보를 기술할 수 있는 프레임워크를 제공하는 것이 그 목적이다. 그러므로 SensorML이 USN 메타데이터를 기술하기 위해서 최적화 되었다고 보기는 어렵다.

본 논문에서는 센서 장치, 센서 노드 그리고 센서 네트워크 정보를 응용 입장에서 바라보고 기술하기 위한 USN 메타데이터를 정의한다. 또한 논문은 제안된 USN 메타데이터를 효과적으로 저장하고 검색하기 위한 방법을 제안하고 이를 기반으로 하는 USN 메타데이터 관리 시스템을 구현한다. 메타데이터 관리 시스템은 성능평가를 통해 그 유효성을 보인다. 논문에서 제안하는 센서 메타데이터는 SensorML을 기반으로 하므로 USN 환경에서 상호운용성을 유지할 수 있을 것이며, 메타데이터 관리 시스템은 USN 미들웨어나 응용에서 메타데이터 관리를 위해 직접 활용될 수 있을 것이다.

▶ Keyword : 유에스엔 메타데이터, 메타데이터 관리 시스템, 유에스엔 미들웨어

### Abstract

The goal of Ubiquitous Sensor Network(USN) environments is to provide users high quality

• 제1저자 : 박종현 • 교신저자 : 강지훈

• 투고일 : 2011. 01. 28, 심사일 : 2011. 02. 18, 게재확정일 : 2011. 03. 03.

\*, \*\* 충남대학교 컴퓨터공학과(Dept. of Computer Engineering, Chungnam National University)

※ 이 연구는 2010년도 충남대학교 학술연구비에 의해 지원되었음.

services based on a variety of sensors. In this environment, sensor devices, sensor nodes and sensor networks are heterogeneous and have various characteristics. Therefore it is important for interoperability to define a metadata for describing USN resources. The OGC(Open Geospatial Consortium) proposes SensorML(Sensor Model Language) as a standard language for modeling sensors. However, SensorML provides a framework for describing a processing model among sensors rather than describing information of sensors. Therefore, to describe a USN metadata is not main purposes of SensorML.

This paper defines a USN metadata which describes information about sensor device, sensor node, and sensor network. Also the paper proposes a method for efficiently storing and searching the USN metadata and implements a USN metadata management system based on our method. We show that our metadata management system is reasonable for managing the USN metadata through performance evaluation. Our USN metadata keeps the interoperability in USN environments because the metadata is designed on SensorML. The USN metadata management system can be used directly for a USN middleware or USN application.

▶ Keyword : USN Metadata, Metadata Management System, USN Middleware

## 1. 서론

센서를 기반으로 사용자에게 다양한 고급의 서비스를 제공하기 위한 유비쿼터스 센서 네트워크 (Ubiquitous Sensor Network: USN) 환경은 이질적인 수많은 자원들로 구성되어 있다. 또한 이러한 USN 자원을 기반으로 사용자에게 다양한 서비스를 제공하기 위한 응용들 역시 그 요구사항이 서로 상이하다. 그러므로 USN 자원들과 센서 기반 응용들 사이의 상호운용을 위하여 센서 자원에 대한 정보를 표준화된 형태로 표현하고 교환하기 위한 메타데이터 표준은 반드시 필요하다.

이러한 요구사항에 발맞추어 OGC(Open Geospatial Consortium)에서는 SensorML(Sensor Model Language)을 제안하여 센서 기반의 처리 모델을 기술하기 위한 표준을 제안하고 있다[1]. 그러나 SensorML의 주목적은 센서를 기반으로 하는 프로세스를 모델링하는 것이고, 이때 필요한 센서들의 정보를 추가적으로 기술할 수 있도록 한다. 그러므로 SensorML 자체를 USN 자원을 기술하기 위한 메타데이터로 사용하기에는 충분하지 못한 부분이 존재한다. 물론 SensorML에서는 사용자가 필요에 따라 부족한 부분을 정의해서 사용할 수 있도록 한다. 그러나 이는 응용이나 도메인에 의존적인 메타데이터이므로 그 사용에 제약이 존재한다. 한편 국내에서는 USN 자원을 기술하기 위한 또 다른 표준안으로 TTA(Telecommunications Technology Association: 한국정보통신기술협회)에서 USN 메타데이터 모델 표준을 제안하고 있다[2]. TTA 표준은 USN 메타데이

터를 기술하기 위하여 제안되었으므로 표현력은 풍부할 수 있으나, 이미 기존에 표준으로 제안된 SensorML과 함께 사용할 경우 많은 부분이 중복으로 정의되어 있으므로 상호 호환에 어려움이 존재한다. 본 논문에서는 이러한 고려사항들을 기반으로 USN 자원을 기술하기 위한 메타데이터를 제안한다. 논문에서 제안하는 메타데이터는 SensorML과의 연동을 고려하여 SensorML을 기반으로 USN 자원을 기술하지만 보다 충실히 USN 자원을 기술하기 위하여 SensorML을 확장한다. 또한 SensorML로 표현할 수 없는 부분은 TTA 표준을 참고하여 확장하므로 두 표준의 장점을 모두 취한다.

USN 메타데이터는 상호운용성을 고려하여 웹상의 데이터 표현을 위한 표준인 XML로 기술된다. 또한 USN 메타데이터는 그 특성상 대용량으로 존재하며, 특별한 경우 동적으로 발생하는 센싱 데이터와 함께 존재한다. 그러므로 이러한 특성을 고려하여 USN 메타데이터를 효과적으로 저장하고 관리하기 위한 방법은 사용자에게 보다 질 좋은 USN 기반 서비스를 제공하기 위해서 반드시 필요하다. 그러나 국내외적으로 아직까지 USN 메타데이터 표준을 연구 중에 있으며, 특별히 USN 메타데이터를 효과적으로 관리하기 위한 연구를 찾아보기는 쉽지 않다[3]. 그러므로 본 논문에서는 USN 메타데이터의 특성을 고려하여 이를 기반으로 메타데이터를 효과적으로 저장하고 관리하기 위한 방법을 제안한다. 제안된 방법은 프로토타입 시스템으로 구현되어 그 성능을 평가한다.

본 논문에서 제안하는 USN 메타데이터는 현재 국내외적으로 활발히 연구 중인 USN 환경에서 USN 자원을 기술하기 위한 메타데이터 표준을 제정하기 위해서 참고 자료로 활

용될 수 있을 것으로 사료된다. 또한 USN 메타데이터 관리 시스템은 USN 미들웨어 혹은 USN 응용에서 메타데이터의 관리를 위해서 직접적으로 사용될 수 있을 것으로 기대된다.

본 논문의 나머지 구성은 다음과 같다. 2절에는 본 논문과 관련된 연구들을 기술하고 있으며, 3절에서는 본 논문에서 제안하는 USN 메타데이터를 설명한다. 4절은 USN 메타데이터 관리 시스템과 메타데이터를 효과적으로 저장하고 관리하기 위한 방법을 제안하며, 5절에서 그 성능을 평가하여 유효성을 보인다. 마지막으로 6절에서 결론과 함께 향후 연구의 진행 방향에 대해서 기술한다.

## II. 관련 연구

센서 네트워크의 발전과 함께 USN 메타데이터에 관한 다양한 연구가 진행되고 있으며[4, 5], 국내외에서 이를 표준화하기 위한 표준안이 제안된바 있다[1, 2]. SensorML[1]은 OGC에서 Sensor Web Enablement(SWE)를 위해 제안한 표준들 가운데 하나로, 센서로부터 측정된 데이터와 센서의 정보를 포함하는 표준 모델과 XML 인코딩을 제공한다. SensorML의 목적은 센서를 기반으로 하는 프로세스들을 모델링하는 것이고, 해당 프로세스를 모델링하기 위해서 필요한 USN 정보를 함께 기술할 수 있도록 하고 있다. 그러므로 SensorML 자체가 USN 메타데이터일 수는 없으나 이를 기반으로 USN 메타데이터를 정의한다면 SensorML 표준과 함께 활용할 수 있으므로 상호운용성 측면에서 매우 유용할 것이다.

국내의 경우, TTA(Telecommunications Technology Association)에서는 자체적으로 USN 메타데이터 모델을 정의하여 사용하도록 권고하고 있다[2]. [2]는 그 목적 자체가 순수하게 USN 메타데이터를 기술하는 것이므로 SensorML에 비해 메타데이터 중심적이며, 정보를 구조적으로 충실하게 기술할 수 있도록 정의되어 있다. [6]에서는 SensorML은 Bottom-Up 접근 방법으로 센서를 기술하고 있으며, TTA에서 제안하는 메타데이터 표준안은 Top-Down 접근 방법으로 센서를 기술하고 있다고 표현하고 있다. 이렇듯 두 표준안은 USN 자원을 기술하기 위한 광의의 목적은 유사하지만 순수하게 USN 메타데이터를 기술하기 위한 측면에서의 접근 방법은 다르다. 특별히 TTA의 표준안은 이미 앞서 표준안이 제안된 SensorML과 독립적으로 제안되었으므로, USN 응용이나 미들웨어 입장에서 이 둘을 함께 사용하면 잉여의 정보가 발생할 수 있다.

본 논문에서는 두 표준안의 특성을 고려하여 SensorML

을 기반으로 하는 USN 메타데이터를 제안하고 있다. 논문에서 제안하는 메타데이터는 USN 자원들을 기술하기 위하여 SensorML에서 부족한 부분을 TTA에서 제안한 메타데이터 표준을 참고하여 확장하며, SensorML로 기술할 수 없는 부분들을 추가로 정의한다. 그러므로 논문에서 제안하는 메타데이터는 SensorML과 상호 연동할 수 있으며, TTA에서 제안한 USN 메타데이터를 모두 지원할 수 있다.

다양한 센서 네트워크 환경에서 SensorML을 기반으로 하는 메타데이터의 필요성과 센서 데이터 및 메타데이터를 효과적으로 저장하고 관리하기 위한 필요성을 언급하고 있는 몇몇 연구들이 존재한다[7, 8, 9]. 그러나 그들은 각각의 응용에서 SensorML의 필요성만 언급하고 있을 뿐 구체적으로 SensorML을 어떻게 사용할 것인가는 다루지 않고 있다. 본 연구에서는 이러한 필요성을 인지하고 실제 SensorML을 기반으로 센서를 위한 메타데이터를 설계하고 이를 효과적으로 저장하고 관리하기 위한 프로토타입 시스템을 개발한다. [10, 11]에서는 본 논문의 목적과 유사하게 SensorML을 기반으로 USN 메타데이터를 기술하고 이를 효과적으로 저장하고 관리하기 위한 시스템을 제안하고 있다. 그러나 [10, 11]에서는 메타데이터를 기술하기 위하여 순수하게 SensorML을 이용했으나 본 논문에서는 이를 확장한 USN 메타데이터를 제안한다. 또한 [10, 11]에서 제안하고 있는 메타데이터 관리 시스템은 특정 관계형 데이터베이스를 기반으로 메타데이터를 저장하고 관리하지만, 본 연구에서는 데이터베이스의 종류에 무관하게 메타데이터를 저장하고 관리할 수 있는 방법을 제안한다. 또한 USN 메타데이터가 XML로 기술된다는 점을 고려하여 메타데이터 검색을 위한 질의어로 XQuery 질의를 이용하여 상호운용성을 보장한다. 더욱이 [10, 11]에서는 메타데이터 관리 모듈의 구조만을 제안하고 있을 뿐 구체적으로 어떻게 구현하였으며 이를 검증하기 위한 성능 평가에 대한 언급은 없다. 그러나 본 논문에서는 구체적으로 메타데이터 관리 시스템을 설계하고 실제 프로토타입을 구현하여 성능 평가를 통한 유효성을 보인다.

## III. USN 메타데이터

본 논문에서 제안하는 USN 메타데이터는 <그림 1>과 같이 크게 세 종류의 메타데이터로 구성되어있다. ‘SensorNetwork’는 센서 네트워크에 대한 메타데이터를 기술하며, ‘System’은 센서를 포함하고 있는 센서 노드를 그리고 ‘Component’는 센서 장치를 기술한다. 각 메타데이터는 개별적으로 기술되며 필요에 따라 상호 참조에 의해서 연결된다. 예를 들면,

'SensorNetwork' 메타데이터는 한 개 이상의 'System' 메타데이터와 연결될 수 있으며, 'System' 메타데이터는 한 개 이상의 'Component' 메타데이터와 연결될 수 있다. 이러한 메타데이터의 구조는 TTA에서 제안하는 USN 메타데이터 모델과 유사해 보인다. 그러나 TTA의 메타데이터는 최상위에 'sensorNetwork' 노드를 두고 그 하위에 'sensorNode' 노드를 자식으로 연결하고 'transducer' 노드를 'sensorNode'의 자식 노드로 연결하였다. 그러므로 센서 네트워크 중심으로 상위에서 하위로 메타데이터를 기술하도록 정의하고 있다. 그러나 본 논문에서는 센서 네트워크 중심으로 메타데이터를 기술하는 것이 아니고, USN 자원을 세 부분으로 분할하여 각각 독립적으로 메타데이터를 기술한다. 즉, 각 메타데이터를 개별적으로 구분하여 기술하고 필요에 따라 참조에 의해 메타데이터들을 연계할 수 있도록 한다. 그러므로 각 노드들 사이에 독립성을 유지하는 한편 다양한 센서 네트워크 환경에 유연하게 대처할 수 있도록 설계하였다. 예를 들어, 특정 노드가 서로 다른 두 센서 네트워크에 동시에 참여할 경우 노드의 중복 선언을 피할 수 있으며, 동적인 네트워크에서 참여하는 경우 수시로 변경되는 네트워크에 유연하게 대처할 수 있다는 장점이 존재한다.

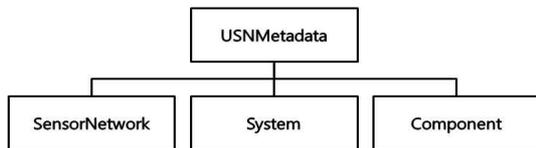


그림 1. USN 메타데이터 구조  
Fig. 1. The Architecture of the USN Metadata

SensorML에서는 물리적 프로세스를 기술하기 위하여 컴포넌트(Component)와 시스템(System)을 정의하고 있다. 이는 TTA에서 제안하는 USN 메타데이터 모델의 센서 노드(Sensor Node)와 트랜스듀서(Transducer)의 개념과 유사하다. 그러나 SensorML은 센서 네트워크를 기술하기 위한 부분을 구분하여 정의하고 있지 않기 때문에 SensorML을 이용하여 센서 네트워크 정보를 기술하기에는 어려움이 따른다. 그러므로 본 논문에서는 센서 네트워크를 기술하기 위하여 <그림 2>와 같은 구조의 센서 네트워크 메타데이터를 제안한다.

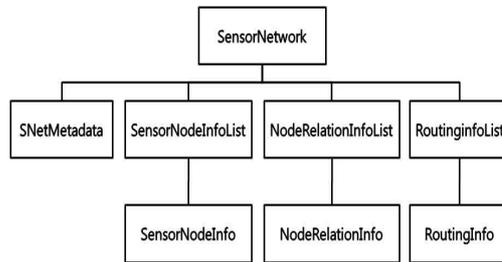


그림 2. SensorNetwork 메타데이터 구조  
Fig. 2. The Architecture of the SensorNetwork Metadata

'SensorNetwork' 메타데이터는 크게 네 부분으로 구성된다. 'SNetMetadata'는 하위에 네트워크의 위치, 능력 등과 같은 센서 네트워크에 관한 메타 정보를 기술한다. 'SensorNodeInfo'는 해당 네트워크에 포함된 센서 노드의 정보를 기술한다. 물론 <그림 1>의 'System' 메타데이터에 센서 노드에 대한 구체적인 정보가 모두 기술되므로 'SensorNetwork' 메타데이터는 이를 참조하여 센서 노드의 정보를 취할 수 있다. 그러나 'SensorNodeInfo'에는 현재 네트워크 내에서 해당 노드의 역할, 위치 등과 같은 네트워크에 의존적인 센서 노드의 정보를 기술한다. 'NodeRelationInfo'는 네트워크 내에 각 노드들 사이의 연결 비용, 이웃 노드의 주소 등과 같은 노드들 사이의 연결 정보를 기술한다. 마지막으로 'RoutingInfo'는 노드들 사이에 라우팅을 위한 정보를 기술한다.

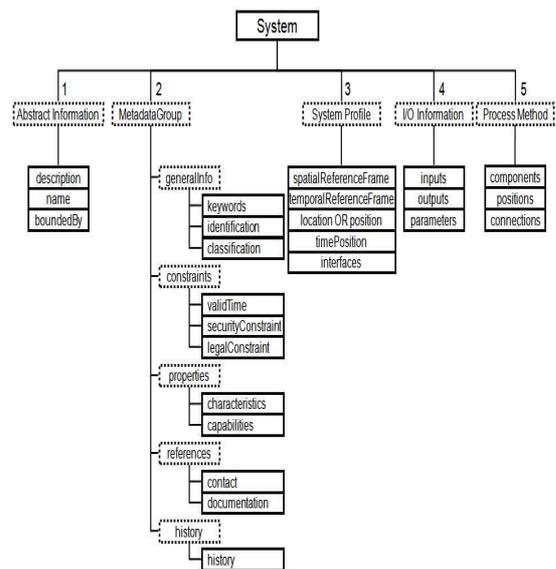


그림 3. System 메타데이터 구조  
Fig. 3. The Architecture of the System Metadata

센서 노드와 센서 장치를 기술하기 위한 'System' 메타데이터와 'Component' 메타데이터는 SensorML의 구조를 참조하여 정의한다. 그러나 앞서 언급한 것처럼 SensorML 자체만으로는 USN 메타데이터를 기술하기 위해 부족한 부분이 존재하므로 본 논문에서는 <그림 3>과 같이 센서 노드를 기술하기 위한 구조를 확장한다.

'System' 메타데이터의 구성은 크게 다섯 부분으로 구성된다. 'Abstract Information'은 센서 노드의 가장 기본 정보들을 기술하며, 'Metadata Group'은 센서 노드의 일반적인 메타 정보를 기술한다. 그림에서는 주요 메타 정보만을 기술하고 있으나 이 부분은 응용의 필요에 따라 SensorML을 기반으로 확장하여 사용할 수 있다. 'System Profile'은 해당 센서 노드의 특성을 기술하고 있으며, 'I/O Information'은 센서 노드의 입출력 정보를 기술한다. 마지막으로 'Process Method'는 해당 센서 노드를 수행하기 위해 필요한 정보와 함께 노드가 포함한 센서 장치의 정보도 필요하다면 함께 기술한다. 센서 장치를 기술하기 위한 'Component' 메타데이터의 기본 구조는 <그림 3>의 'System' 메타데이터와 크게 다르지 않다. 단지 'Process Method' 부분에 센서 장치를 참조하는 부분이 존재하지 않고 몇몇 정보들의 속성이 필수 요소 혹은 선택 요소로 변경되었을 뿐이다.

#### IV. USN 메타데이터 저장 관리 시스템

XML은 인터넷 상에 데이터 교환을 위한 표준이다. 그러므로 상호운용성을 위하여 USN 환경에서 메타데이터 교환을 위하여 XML을 사용하는 것은 자연스러운 선택이다. 이러한 이유에서 본 논문에서 제안하고 있는 USN 메타데이터를 비롯하여 SensorML과 TTA의 USN 메타데이터 모델 모두 XML로 기술된다. 그러므로 결국 USN 메타데이터를 관리하는 것은 XML 데이터를 관리하는 것과 직결된다.

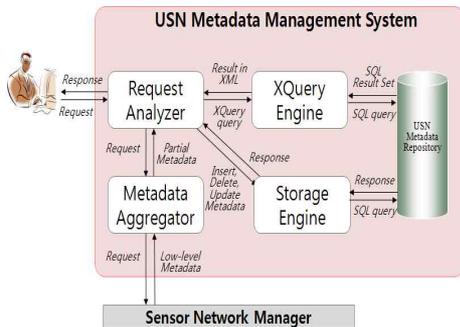


그림 4. USN 메타데이터 관리 시스템  
Fig. 4. The USN Metadata Management System

<그림 4>는 USN 메타데이터 관리 시스템의 구조를 보인다. USN 메타데이터 관리 시스템은 크게 네 개의 모듈로 구성된다. 'Request Analyzer'는 사용자의 요구사항을 분석하고 요구에 따라 적절한 모듈을 호출하는 역할을 담당한다. 'Storage Engine'은 USN 메타데이터의 삽입, 삭제, 갱신을 수행하며, 'XQuery Engine'은 저장소에 저장된 메타데이터의 검색을 수행한다. 'Metadata Aggregator'는 사용자가 메타데이터 삽입을 요청했을 경우, 필요에 따라 센서 네트워크로부터 직접 센서 장치, 센서 노드, 그리고 센서 네트워크의 정보를 취하는 역할을 담당한다. 'Sensor Network Manager'는 센서 네트워크의 싱크 노드(Sink Node)와 통신하여 센싱 데이터 및 USN 메타데이터를 수집하여 USN 메타데이터 관리 시스템에 반환한다. 물론 사용자가 모든 USN 메타데이터를 입력할 수 있다면 이러한 과정은 생략될 수 있다. 그러나 현실적으로 USN 메타데이터 가운데에는 노드의 전원 잔량과 같은 동적으로 변하는 정보들이 존재하므로 이러한 접근 방법이 요구된다.

##### 1. USN 메타데이터 저장 엔진

USN 메타데이터는 XML로 기술되며 그 양이 대용량이다. 그러므로 본 논문에서는 이를 저장하기 위한 저장소로 대용량 데이터를 효과적으로 관리하기 위해서 그 성능이 이미 검증된 관계형 데이터베이스를 사용한다. 물론 XML 데이터를 효과적으로 관리하기 위한 XML 전용 데이터베이스도 존재한다. 그러나 관계형 데이터베이스를 선택한 이유는 센서 데이터와 메타데이터를 동시에 효과적으로 관리할 수 있도록 하기 위함이다. 현재 대부분의 상용 관계형 데이터베이스는 XML을 지원한다. 그러나 아직까지 XML을 지원하기 위한 표준 SQL은 존재하지 않으므로 상용 관계형 데이터베이스들은 XML을 지원하기 위해서 내부적으로 정의한 XML 지원 방법을 사용한다. 이러한 현실은 USN 메타데이터 관리 시스템들이 서로 다른 관계형 데이터베이스를 사용할 경우 그에 의존적인 저장과 검색 방법을 사용해야한다는 단점이 존재한다. 그러므로 본 논문에서는 관계형 데이터베이스의 종류에 무관하게 USN 메타데이터를 저장하고 관리하기 위한 방법을 제안한다.

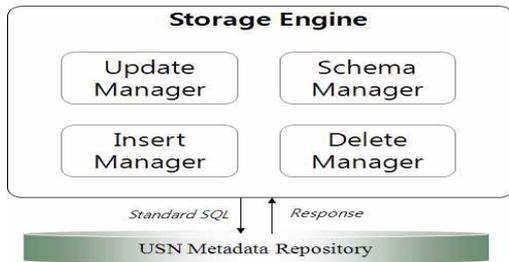


그림 5. 저장 엔진의 구조  
Fig. 5. The Architecture of the Storage Engine

<그림 5>는 USN 메타데이터를 위한 저장 엔진의 모습을 보인다. 'Insert Manager', 'Delete Manager', 'Update Manager'는 각각 메타데이터를 저장, 삭제, 그리고 갱신하는 역할을 담당한다. 'Schema Manager'는 시스템 관리자가 데이터베이스 스키마를 관리하기 위해서 사용된다.

본 논문의 앞선 연구로 우리는 디지털 방송을 위한 표준 메타데이터인 TV-Anytime 메타데이터를 방송용 사용자 단말(Settop-Box)에서 효과적으로 관리하기 위한 시스템을 개발하고 그 성능을 보인바 있다[12]. TV-Anytime 메타데이터 역시 XML로 기술된다. 그러나 USN 메타데이터 관리 시스템과는 달리 방송용 사용자 단말의 하드웨어 및 소프트웨어 자원은 제한되어 있으므로 이를 효과적으로 관리하기 위하여 데이터베이스를 사용하지 않고 파일 기반으로 메타데이터를 관리하였다. 그러나 그 접근 방법은 관계형 데이터베이스를 기반으로 활용할 수 있다.

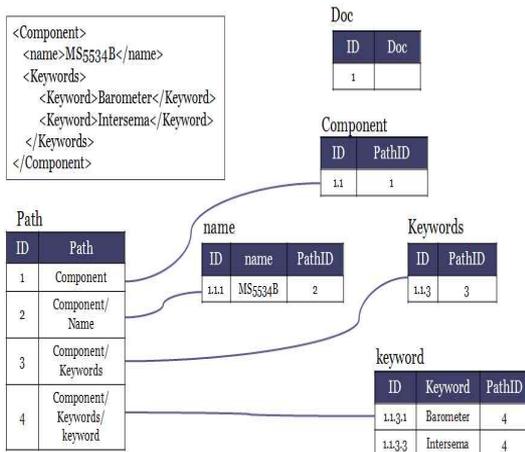


그림 6. USN 메타데이터 저장을 위한 관계형 테이블 구조  
Fig. 6. The Relational Table Structure for USN Metadata

<그림 6>은 USN 메타데이터를 관계형 데이터베이스에 저

장하기 위한 관계형 테이블 구조를 보인다. XML 데이터를 관계형 데이터베이스에 저장하기 위한 가장 기본적인 방법은 [13]에서 제안된 것처럼 모든 노드들을 하나의 테이블에 저장하거나 (Universal Approach), 각 노드마다 하나의 테이블을 생성하여 저장하는 방법(Binary Approach)이다. 두 방법 모두 장점과 단점이 존재하지만 본 논문에서는 각 노드마다 하나의 테이블을 생성하는 Binary 방법을 사용한다. 즉, 데이터베이스에 생성되는 테이블의 수는 USN 메타데이터에 존재하는 노드 종류의 수만큼 발생할 것이다. 그러나 USN 메타데이터는 단일의 스키마를 기반으로 작성되므로 노드의 수는 제한되어 있다. 그러므로 이를 사용할 경우 생성되는 테이블의 수는 유한개이다. Binary 방법의 장점은 노드 단위로 테이블을 생성하므로 Universal 방법에 비해서 저장소 낭비를 줄일 수 있다는 점이지만 가장 큰 단점은 특정 노드의 경로를 탐색할 경우(XPath 처리 시) 경로에 포함된 모든 테이블을 조인해야한다는 것이다. 그러므로 이를 해결하기 위하여 본 논문에서는 Path 테이블[14]을 사용한다. Path 테이블은 XML 문서에 존재하는 모든 경로를 테이블에 저장하였다가 사용자의 XPath 질의가 요청되었을 경우 테이블에 저장된 값과 비교하여 해당 경로에 일치하는 테이블을 반환하는 방법이다. 이를 사용한다면 XPath 처리를 위하여 테이블 사이의 조인 없이 단순히 스트링 비교만으로 직접 해당 테이블에 접근할 수 있다는 장점이 존재한다. 이 또한 USN 메타데이터가 단일의 스키마를 따르기 때문에 사용할 수 있다. 즉, 단일의 스키마를 기반으로 작성된 USN 메타데이터에 존재하는 노드의 경로는 유한개이다. 물론 USN 메타데이터에도 자손 노드가 조상 노드를 자식 노드로 갖거나 스스로를 자식 노드로 갖는 경우도 존재한다. 이러한 경우 최상위 노드에서 해당 노드까지의 경로는 무한정 증가할 수 있지만, 실제 USN 메타데이터 문서에서 대부분의 경우 그 경로는 일정하며 패턴 역시 유사하다. 실제 본 논문에서 성능 평가를 위하여 시스템을 구현하고 견본 USN 메타데이터를 관계형 데이터베이스에 저장한 결과 각 노드를 지정(Addressing)하는 경로의 수는 300개를 넘지 않았다. Path 테이블을 이용하여 XQuery 질의에 기술된 특정 노드를 위한 XPath 경로를 찾을 수 있다할 지라도, 올바른 질의 처리를 위해서는 XQuery 질의에 존재하는 노드들 사이의 연관 관계를 알아야한다. 예를 들어, 경로 "A/B/C"를 갖는 노드 C와 "A/B/D"의 경로를 갖는 노드 D는 부모 노드인 B 노드가 동일하다면 형제 관계일 것이고, 서로 다른 B 노드가 부모 노드라면 형제 관계가 아니다. 그러므로 본 논문에서는 메타데이터에 존재하는 모든 노드들 사이의 관계를 표현하기 위하여 [15]에서 제안한 ORDPATH 방

법을 이용하여 각 노드에 ID를 부여한다. ORDPATH 방법은 문서의 최상위 노드부터 해당 노드까지의 경로를 10진 분류법을 이용하여 나타내는 방법으로, 모든 노드의 ID가 될 수 있을 뿐만 아니라 부모-자식의 연결, 형제 노드들의 순서 등과 같은 각 노드들의 연관 관계를 알 수 있다. 이러한 특성은 기존의 Binary 방법에서 노드의 구조를 표현하기 위해서 사용된 추가적인 정보(부모-자식 간의 정보, 형제들 간의 정보)들을 제거할 수 있도록 하므로 저장 공간을 절약할 수 있다는 장점을 함께 갖는다.

본 논문에서는 USN 메타데이터를 관계형 데이터베이스에 저장할 때 저장 공간의 낭비를 줄이고 검색의 효율을 높이기 위해 기존에 XML 데이터를 관리하기 위해 제안된 다양한 방법들을 혼합하여 사용한다. 이때 한 가지 방법의 장점만을 취하고 단점은 다른 방법의 장점으로 제거한다. 이러한 접근 방법은 USN 메타데이터의 특성을 고려하였으므로 가능한 일이다.

### 2. XQuery 엔진

USN 메타데이터는 XML로 기술되므로 XML 데이터 검색을 위한 표준 질의어인 XQuery를 USN 메타데이터 검색을 위해 사용하는 것은 매우 자연스러운 접근 방법이다. 그러나 USN 메타데이터는 관계형 데이터베이스에 테이블 형태로 저장되어 있으므로 XQuery를 직접 사용하여 데이터를 검색할 수는 없다. 즉, 사용자가 질의한 XQuery 질의를 처리하기 위해서는 본 논문에서 제안한 관계형 테이블 구조에 맞는 SQL 질의로 변환하여 처리해야한다. 또한 그 결과인 SQL 결과 집합(Result Set) 역시 XQuery 질의의 결과인 XML 데이터로 변환해서 반환해야한다.

<그림 7>은 XQuery 엔진의 구조를 보인다. 사용자가 질의한 XQuery 질의는 'XQuery Analyzer'에 의해서 구문 분석된다. XQuery 질의는 메타데이터를 검색하기 위한 부분과 결과를 생성하기 위한 반환 부분으로 구분된다. 구문 분석된 XQuery 질의 가운데 메타데이터를 검색하기 위한 부분은 'SQL Generator'에 의해서 관계형 데이터베이스에 질의하기 위한 SQL 질의로 변환된다. 'Query Executor'는 변환된 SQL 질의를 실제 메타데이터가 저장된 관계형 데이터베이스에 질의하여 SQL 결과를 얻는다. 이렇게 얻어진 SQL 결과는 'XML Publisher'에 의해서 XML 문서로 변환된다. 마지막으로 'XQuery Analyzer'는 앞서 구문 분석 시 추출된 반환 구조를 반영하여 사용자에게 반환하기 위한 USN 메타데이터를 생성한다.

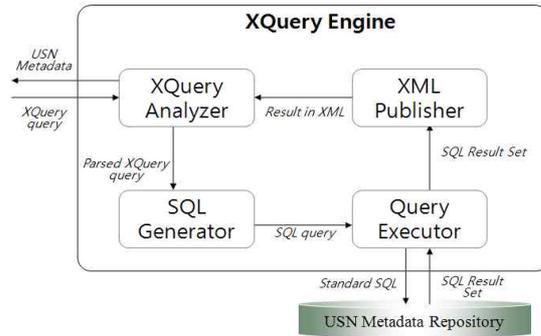


그림 7. XQuery 엔진의 구조  
Fig. 7. The Architecture of XQuery Engine

<그림 8>은 XQuery 질의와 변환된 SQL 질의를 보인다. XQuery 질의의 내용은 'sml:System'을 태그 명으로 가지는 엘리먼트 노드들 가운데 'gml:id' 속성 값이 'SN120\_0001'인 USN 메타데이터 문서 전체를 반환하라는 것이다. 이를 SQL 질의로 변환하기 위해서 'sml:System'이라는 이름의 노드가 저장된 테이블 'sml\_System'에서 id 속성의 값이 'SN120\_0001'인 레코드를 모두 취한다. 이때 각 레코드들은 Path 테이블의 값이 '/sml:SensorML/sml:member/sml:System'인 레코드와 조인되어야한다.

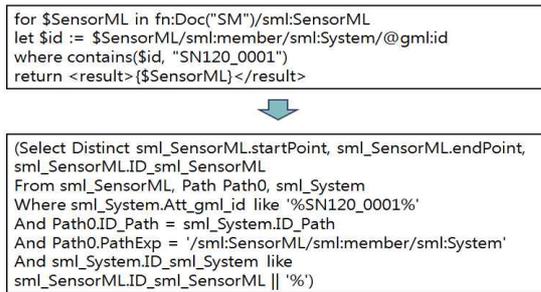


그림 8. XQuery 질의와 변환된 SQL 질의  
Fig. 8. XQuery query and Translated SQL query

또한 올바른 경로를 찾기 위하여 'sml\_System' 테이블로부터 선택된 레코드의 ID가 'sml\_SensorML' 테이블의 어떤 레코드 ID로 부터 파생되었는지를 검색하여 올바른 부모 노드를 찾는다.

### 3. 사용자 인터페이스

사용자의 사용 편의를 제공하기 위하여 본 논문에서는 <그림 9>와 같은 USN 메타데이터 관리 시스템의 사용자 인터페이스를 제공한다. <그림 10>은 USN 메타데이터를 생성하기

위한 인터페이스이다. 화면 좌측의 트리 구조는 USN 메타데이터에 포함된 노드들의 구조적으로 표현한다. 사용자는 입력을 원하는 노드를 클릭하여 메타데이터를 입력할 수 있다. 화면 우측은 생성된 USN 메타데이터를 보이며 필요에 따라 직접 편집이 가능하다.

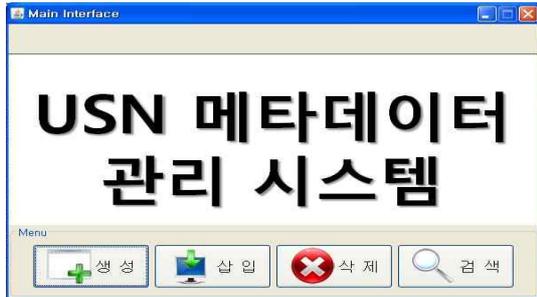


그림 9. USN 메타데이터 관리 시스템 메인 인터페이스  
Fig. 9. The Main Interface for USN Metadata Management System

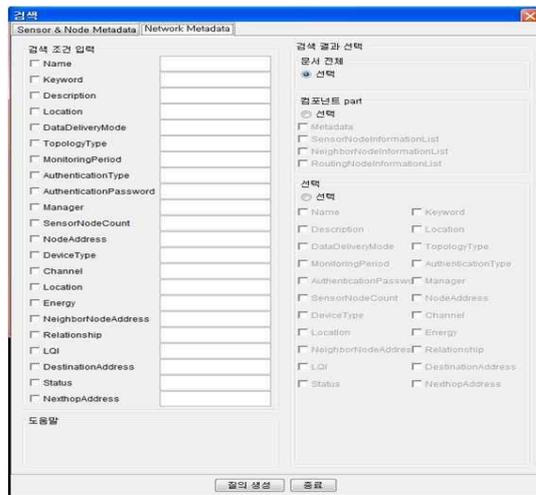


그림 11. XQuery 질의 생성을 위한 인터페이스  
Fig. 11. The Interface for generating an XQuery query

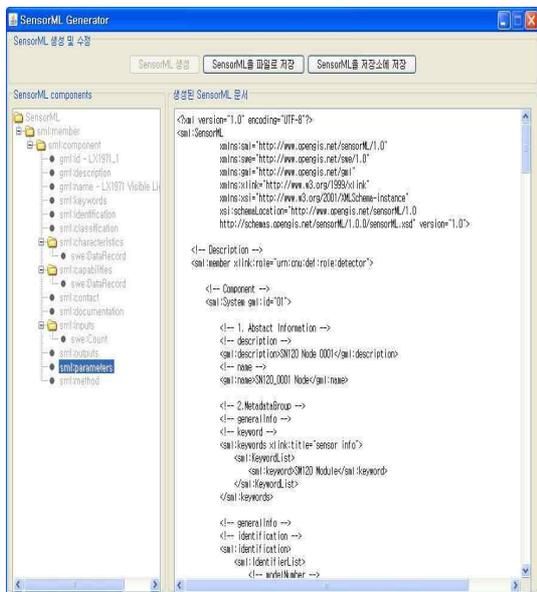


그림 10. USN 메타데이터 생성 인터페이스  
Fig. 10. The Interface for generating a USN Metadata instance

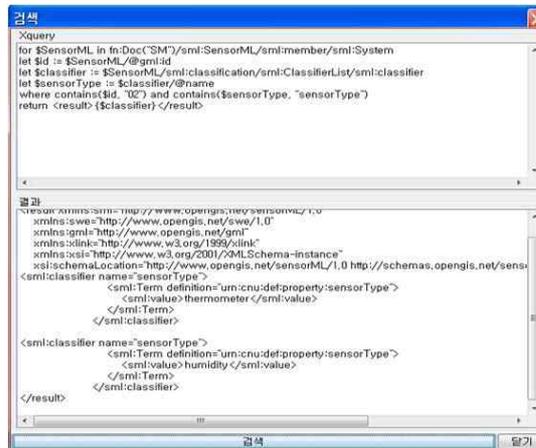


그림 12. USN 메타데이터 검색 결과  
Fig. 12. The Result of USN Metadata Search

XQuery는 표현력이 풍부하다는 장점과 함께 일반 사용자가 사용하기에는 복잡하다는 단점이 존재한다. 그러므로 본 논문에서는 XQuery 질의를 생성하기 위한 사용자 인터페이스를 제공한다. <그림 11>의 좌측은 사용자가 검색을 위해 자주 사용하는 조건들이고 우측은 반환 조건들이다. 물론 사용자가 XQuery에 익숙하다면 직접 질의를 생성할 수도 있으며, 사용자 인터페이스를 이용해 기본 질의 형식을 생성한 후 편집할 수도 있다. <그림 12>의 상단은 XQuery 질의이며, 하단은 이를 수행하여 USN 메타데이터를 추출한 모습을 보인다.

### V. 성능 평가

USN 메타데이터 관리 시스템의 성능을 평가하기 위하여 본 논문에서는 Intel Pentium IV 3.2GHz CPU와 2GB 메모리의 시스템을 사용하였다. 시스템의 운영체제는 Windows XP이고, 데이터베이스는 Oracle Enterprise Manager 10g production (10.2.0.2.1)을 사용하였다. 성능 평가는 Oracle 10g 자체적으로 지원하는 XQuery 처리와 본 논문에서 제안하는 XQuery 질의 처리의 수행 시간을 측정하는 방법을 사용했다.

표 1. 성능 평가를 위한 건본 XQuery 질의들의 특징  
Table 1. The characteristic of sample XQuery queries for performance evaluation

Query	조건(Condition)	반환(Return)
XQ1	단일 조건	단일 노드
XQ2	단일 조건	메타데이터 전체
XQ3	조건 3개	메타데이터 전체
XQ4	조건 5개	메타데이터 전체
XQ5	조건 7개	메타데이터 전체

<표 1>은 성능 평가를 위한 XQuery 질의들의 특성을 보인다. XQ1은 하나의 조건으로 단일의 노드를 반환하기 위한 질의이고, XQ2는 <그림 8>에 기술된 XQuery 질의와 같이 단일 조건으로 메타데이터 문서 전체를 반환하기 위한 질의이다. XQ6은 Where절에 7개의 조건을 포함하며 이를 모두 만족하는 USN 메타데이터 문서 전체를 취하는 질의이다.

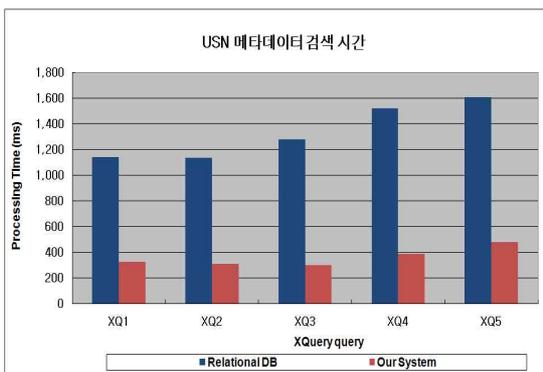


그림 13. USN 메타데이터 검색 시간  
Fig. 13. Searching time of USN Metadata

<그림 13>은 약 5MB의 USN 메타데이터를 대상으로 <표

1>에 기술된 5개의 XQuery 질의를 수행한 검색 시간을 보인다. 그림에서 알 수 있는 것처럼 본 논문에서 제안한 방법은 질의의 종류에 무관하게 USN 메타데이터를 검색하기 위해서 우수한 성능을 보인다.

<그림 14>는 USN 메타데이터의 크기를 5MB에서 20MB까지 5MB씩 증가시키면서 XQ5 질의를 수행한 시간을 측정하는 결과이다. 그림에서 알 수 있는 것처럼 본 논문에서 제안한 방법은 데이터의 크기에 독립적으로 우수한 성능을 보인다. 이는 아마도 Path 테이블을 이용하여 직접 특정 노드에 접근이 가능하기 때문일 것이다.

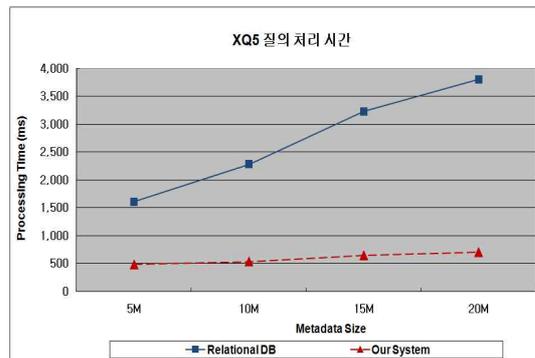


그림 14. 데이터 크기에 따른 성능 평가  
Fig. 14. Performance evaluation for scalability property

성능 평가에 사용된 관계형 데이터베이스는 USN 메타데이터 스키마의 등록이나, 검색을 위한 어떠한 인덱스도 사용하지 않고 가장 기본적인 저장 방법으로 USN 메타데이터를 저장하고 XQuery 질의를 수행시킨 결과이다. 그러므로 두 방법을 직접 비교하는 것은 무리가 있다. 그러나 본 논문에서는 표준 SQL 질의를 사용하므로 관계형 데이터베이스의 종류에 무관하게 메타데이터를 저장하고 검색할 수 있는 방법을 제안하였다. 실제 본 논문에서는 또 다른 상용 데이터베이스인 MS-SQL Server 2005에서도 동일한 실험을 수행했으며, 위 실험 결과와 유사한 결과를 얻을 수 있었다.

### VI. 결론

본 논문은 유비쿼터스 센서 네트워크 환경에서 자원을 기술하기 위한 USN 메타데이터를 제안하였다. 이는 현재 국내에서 제안하고 있는 USN 메타데이터 표준과 국외에서 제안하는 표준을 함께 반영하고 있다. 또한 논문은 USN 메타데이터를 효과적으로 저장하고 관리하기 위하여 USN 메타데이

터 관리 시스템을 제안하였다. 메타데이터 관리 시스템은 효율적인 메타데이터의 저장과 검색을 위하여 기존에 XML 데이터를 관리하기 위해 제안된 여러 방법들의 장점만을 취하고 단점을 제거한 새로운 방법을 사용하고 있다. 이는 USN 메타데이터의 특성을 고려하였기 때문에 가능하다. 또한 논문은 USN 메타데이터 관리 시스템을 일반 사용자가 쉽게 사용할 수 있도록 사용자 인터페이스를 제공하였으며, 메타데이터 관리 시스템 프로토타입을 구현하여 그 성능을 평가했다.

본 논문의 결과는 향후 센서 네트워크 환경에서 USN 자원을 기술하기 위한 국제 표준 메타데이터를 정의하기 위해서 참조될 수 있을 것이다. 또한 USN 메타데이터 관리 시스템은 응용이나 환경에 독립적으로 설계되었으므로, 향후 USN 미들웨어나 USN 응용에서 메타데이터 관리를 위해 직접 사용이 가능 할 것으로 기대된다.

## 참고문헌

- [1] OGC Sensor Model Language, OpenGIS Standard 2007. <http://www.opengeospatial.org/standards/sensorml>
- [2] USN Metadata Model, TTAK.KO-06.0168/R1, June 2009. <http://www.tta.or.kr>
- [3] S. J. Lww, Y. W. Kim, S. K. You, H. J. Kim and H. K. Jung, "Sensor Data Processing System for USN Application System" Proc. of The Korea Institute of Maritime Information and Communication Sciences, 2007.
- [4] Y. Obashi, T. Kokogawa, Y. Zheng, H. Chen, H. Mineno, and T. Mizuno, "A Meta-Data-Based Data Aggregation Scheme in Clustering Wireless Sensor Network", Proc. KES 2007, pp.477-483, September 2007.
- [5] D. Ballari, M. Wachowicz, and M. A. M. Callejo, "Metadata behind the Interoperability of Wireless Sensor Networks", Sensors, Vol.9, no.5, pp.3635-3651, May 2009.
- [6] M. R. Kim, J. W. LEE, Y. J. LEE, and J. C. RYOU, "COSMOS : A Middleware for Integrated Data Processing over Heterogeneous Sensor Networks", ETRI journal, vol. 30, no.5, pp. 696-706, 2008.
- [7] R. Peng, K. A. Hua, and G. L. Hamza-Lup, "A Web Services Environment for Internet-Scale Sensor Computing" Proc. IEEE SCC 2004, China, September 2004.
- [8] G. Aloisio, D. Conte, C. Elefante, I. Epicoco, G. P. Marra, G. Mastrantonio, and G. Quarta, "SensorML for Grid Sensor Networks" Proc. GCA'06, Nevada, USA, June 2006.
- [9] K. Aberer, M. Hauswirth, and A. Salehi, "Infrastructure for data processing in large-scale interconnected sensor networks" Proc. MMDM'07, Germany, May 2007.
- [10] Y. K. Lee, Y. J. Jung, K. H. Ryu, and K. D. Kim, "Design and Implementation of Metadata Model for Sensor Data Stream", Proc. ISRS 2006 PORSEC, Vol.II, pp.768-771, November 2006.
- [11] Y. K. Lee and J. S. Song, "Metadata Management on Ubiquitous Sensor network Environment", Proc. of Korea Spatial Information System Society, 2006.
- [12] J.-H. Park and J.-H. Kang, "TV-Anytime Metadata Management System based on a Set-Top Box for Digital Broadcasting" Journal of The Korea Society of Computer and Information, Vol. 13, No. 4, pp.71-78, July. 2008.
- [13] D. Florescu & D. Kossmann, "Storing and Querying XML Data Using an RDBMS," IEEE Data Engineering Bulletin, Vol. 22, No. 3, September 1999.
- [14] MYoshikawa, T.Aragasa, T.Shimura, & S.Uemura: "XRel: a path-based approach to storage and retrieval of XML documents using relational databases," ACM Transactions on Internet Technology, Vol. 5, August, 2001.
- [15] Patrick O'Neil, Elizabeth O'Neil, Shankar Pal, Istvan Cseri, Gideon Schaller, Nigel Westbury "ORDPATHS: Insert-Friendly XML Node Labels" Proc. ICMD 2004 pp.903-908, 2004.

## 저 자 소 개



### 박 종 현

2002년 충남대학교 컴퓨터과학과 석사졸업.

2007년 충남대학교 컴퓨터과학과 박사졸업.

2007년~2008년 충남대학교 소프트웨어연구소 전임연구원

2009년~2009년 거제대학 조선정보계열 초빙교수

2009년~2010년 큐슈대학교 정보기반연구센터 방문연구원

2011년~현재 충남대학교 컴퓨터공학과 초빙교수

주관심분야 : 상황인지, 추론시스템, XML, Ontology, 유비쿼터스 컴퓨팅,

Email : jonghyunpark@cnu.ac.kr



### 강 지 훈

1981년 한국과학기술원 전산학과 석사졸업.

1996년 한국과학기술원 전산학과 박사졸업.

2000년~2002년 충남대학교 정보통신원장.

1985년~현재 충남대학교 전기정보통신공학부 교수.

주관심분야 : 시맨틱웹, 추론, XML, XQuery, 데이터베이스 시스템, 웹정보시스템.

Email : jhkang@cnu.ac.kr