

Design and Implementation of Sensor Registry Data Model for IoT Environment

Sukhoon Lee[†] · Dongwon Jeong^{**} · Hyunjun Jung^{***} · Doo-Kwon Baik^{****}

ABSTRACT

With emerging the Internet of Things (IoT) paradigm, the sensor network and sensor platform technologies have been changed according to exploding amount of sensors. Sensor Registry System (SRS) as a sensor platform is a system that registers and manages sensor metadata for consistent semantic interpretation in heterogeneous sensor networks. However, the SRS is unsuitable for the IoT environment. Therefore, this paper proposes sensor registry data model to register and manager sensor information in the IoT environment. We analyze Semantic Sensor Network Ontology (SSNO) for improving the existed SRS, and design metamodel based on the analysis result. We also build tables in a relational database using the designed metamodel, then implement SRS as a web application. This paper applies the SSNO and sensor ontology examples with translating into the proposed model in order to verify the suitability of the proposed sensor registry data model. As the evaluation result, the proposed model shows abundant expression of semantics by comparison with existed models.

Keywords : Internet of Things, Sensor Registry System, Sensor Registry Data Model, Sensor Model, Semantic Sensor Network Ontology

IoT 환경을 위한 센서 레지스트리 데이터 모델의 설계 및 구현

이 석 훈[†] · 정 동 원^{**} · 정 현 준^{***} · 백 두 권^{****}

요 약

사물인터넷(Internet of Things, IoT) 패러다임이 대두되며 센서의 개체수가 폭발적으로 증가할 것으로 예상됨에 따라 센서 네트워크 및 센서 플랫폼 기술들이 변화되고 있다. 센서 플랫폼 중 하나인 센서 레지스트리 시스템(Sensor Registry System, SRS)은 이기종 센서 네트워크 환경에서 센서 데이터의 일관성 있는 의미 해석을 위하여 센서 메타데이터를 등록하고 관리하는 시스템이다. 하지만 기존의 SRS는 IoT 환경에 적합한 데이터 구조를 지니고 있지 않다. 따라서 이 논문은 IoT 환경에서 센서 정보들을 관리하고 등록하기 위하여 센서 레지스트리 데이터 모델을 제안한다. 기존의 SRS를 개선하기 위하여 시맨틱 센서 네트워크 온톨로지(Semantic Sensor Network Ontology, SSNO)를 분석하고, 이에 기반한 메타모델을 설계한다. 또한 설계한 메타모델을 이용하여 관계형 데이터베이스의 테이블로 구축하고 SRS를 웹 애플리케이션으로 구현한다. 이 논문은 제안하는 센서 레지스트리 데이터 모델의 적합성을 검증하기 위하여 SSNO 및 센서 온톨로지 예제들을 변환하여 제안 모델에 적용한다. 평가 결과 제안 모델이 기존 연구들보다 더 풍부한 의미 표현이 가능함을 보인다.

키워드 : 사물인터넷, 센서 레지스트리 시스템, 센서 레지스트리 데이터 모델, 센서 모델, 시맨틱 센서 네트워크 온톨로지

1. 서 론

최근 사물인터넷 (Internet of Things, IoT) 패러다임이

미래 기술의 핵심 키워드로 떠오르고 있다[1]. 이에 따라 무선 센서 네트워크(Wireless Sensor Network), 데이터 저장 및 분석 등과 같은 영역에서도 큰 변화가 일어나고 있다[2]. 기존의 대규모 센서 네트워크에 비해 IoT 환경에서 개체(Thing) 내의 센서 네트워크 규모는 작아졌으며, 싱크노드(Sink Node), 센서노드(Sensor Node), 액추에이터(Actuator) 등과 같은 역할들의 구분은 희미해졌다. 반면, IoT에서의 디바이스 종류는 다양해졌으며, 이에 따른 플랫폼 및 데이터 종류와 의미 역시 풍부해졌다. 따라서 기존의 센서 네트워크 영역의 플랫폼에서도 IoT 환경에 적용할 수 있도록 개선할 필요가 있다.

* 이 논문은 2016년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업(NRF-2014R1A1A2058992).

** 이 논문은 2015년도 한국정보처리학회 추계학술발표대회에서 'IoT 환경을 위한 SRS 데이터 모델 설계 및 구현'의 제목으로 발표된 논문을 확장한 것임.

† 준 회 원 : 아주대학교 의료정보학과 연구장사

** 종신회원 : 군산대학교 통계컴퓨터학과 교수

*** 준 회 원 : 고려대학교 컴퓨터·전파통신공학과 박사과정

**** 종신회원 : 고려대학교 컴퓨터·전파통신공학과 교수

Manuscript Received : December 28, 2015

First Revision : April 11, 2016

Accepted : April 11, 2016

* Corresponding Author : Dongwon Jeong(djeong@kunsan.ac.kr),
Doo-Kwon Baik(baikdk@korea.ac.kr)

한편, Sensor Web Enablement (SWE)[3], SensorMap[4], Sensorpedia[5]와 센서 웹 플랫폼들도 다양한 환경과 센서 데이터 제공을 위해 개선되어 왔다. 특히 Sensing as a Service (SenaaS)는 웹이나 클라우드 서비스 등을 통하여 센서들로부터 센싱된 데이터를 제공해 주기 위한 서비스로 일종의 많은 연구들이 미들웨어 형태로 개발되어 왔다[6]. Sensor-Cloud[7], CA4IOT[8]와 같은 연구들은 SenaaS로서, 각각 클라우드 컴퓨팅이나 사용자 문맥 인지 등과 같은 기술을 접목한다. 그중, Sensor Registry System (SRS)은 센서들의 메타데이터를 등록하고 관리하는 시스템으로, 이기종 센서 네트워크 시스템에서 센서 데이터의 일관성 있는 의미 해석을 가능하게 한다[9]. 하지만 기존의 SRS는 센서 메타데이터의 구조가 센서와 그 의미 자체에만 집중되어 있어 앞서 기술한 IoT 환경을 충분히 반영하지 못한다.

따라서 이 논문은 기존의 센서 레지스트리(SR) 데이터 모델을 개선하여 확장된 SR 데이터 모델을 제안한다. IoT 환경을 위하여 W3C의 시맨틱 센서 네트워크 온톨로지(Semantic Sensor Network Ontology, SSNO)[10]를 반영하여 SR 데이터 모델을 설계한다. 이를 위하여, 이 논문은 먼저 SRS의 목적과 요구사항을 분석하고, 이에 따른 유스케이스를 기술한다. 이러한 요구사항과 유스케이스에 기반하여 SSNO에서 반영할 요소와 제외할 요소를 결정하고, 이를 메타모델 및 관계형 데이터베이스(RDB) 테이블로 구축한다. 구축된 데이터 모델을 기반으로 실제 웹 어플리케이션을 구현한다. 마지막으로 SSNO와 센서 온톨로지 예제들을 제안하는 SR 데이터 모델로 적용해 봄으로써 적합성 평가를 수행한다.

제안하는 SR 데이터 모델은 IoT 환경에서 사용자가 사물, 즉 시스템에 접근하여 시스템이 지니는 데이터를 제공받고 그 의미를 이해하고자 할 때, 등록된 센서 메타데이터를 제공해줌으로써 일관성 있는 의미 해석을 가능하게 한다.

이 논문의 구성은 다음과 같다. 2장에서는 관련 연구를 소개하고, 3장에서는 SRS의 요구사항, 목적, 그리고 유스케이스를 기술한다. 4장에서는 이 논문이 제안하는 SR 데이터 모델을 메타모델과 RDB 테이블로 모델링 한다. 5장은 제안 데이터 모델을 이용한 실제 구현 결과를 보이고, 6장에서는 SR 데이터 모델의 적합성을 각 데이터 모델과 비교 평가한다. 마지막으로 7장에서 결론 및 향후 연구에 대하여 서술한다.

2. 관련 연구

이 장은 센서에 관련된 모델 및 온톨로지에 대한 관련 연구를 기술한다. 기존의 센서 정보를 관리하고 제공해 주기 위한 센서 플랫폼들에서도 센서를 정의하기 위하여 센서의 메타모델 및 온톨로지들을 연구하였다[9-13].

기존 SRS에서의 데이터 모델은 센서 메타모델의 의미처리 등을 위하여 센서, 위치 정보, 센서 타입, 측정 단위 등 의미 해석을 위한 최소한의 정보만을 관리한다[9]. 하지만 이러한 이유로 IoT 환경에서의 사용자가 센서에 접근하기 위한 시스템 및 통신 사양과 같은 정보는 포함하지 않는다.

SensorML은 Open Geospatial Consortium (OGC)에서 명세한 표준으로 SWE에 이용하기 위한 센서를 메타모델로 명세한다[11]. 하지만 SensorML은 센서 웹에만 초점을 두고 있어 IoT 환경에 적합하지 않으며 의미 정보를 담을 수 없는 문제를 지닌다.

SSNO는 W3C에서 정의한 시맨틱 센서 네트워크에 대한 온톨로지로서 다양한 영역에서의 온톨로지를 조사하고 통합하여 명세한다[12]. 특히 SSNO는 센서, 관측, 시스템, 특징 및 속성과 같은 4가지 관점에서 센서 온톨로지를 정의한다[13]. 또한, 자극-센서-관측 패턴을 이용하여 기본 구성을 이루고 있다. 하지만 많은 개념을 한 모델에 담고자 하므로 이중 상속과 같은 복잡한 구조와 중복 정의와 같은 애매모호한 개념을 지닌다.

한편, IoT의 관점에서 센서를 모델링한 연구들도 존재한다[14-17]. IoT-A는 European Lighthouse Integrated Project의 일환으로 IoT 아키텍처를 명세한다[14]. IoT-A는 각종 참조 모델과 그에 따른 표준들을 정의한다[15]. 하지만 이러한 참조 모델에도 센서나 액추에이터와 같은 중요한 요소는 개념 혹은 클래스로서 주 모델에 들어있을 뿐, 센서에 대한 구체적 속성이나 요소는 명세하지 않는다.

[16]은 IoT 환경에서 스마트 공간(Smart Space)의 센서와 액추에이터의 기능을 표현하고 자원 지향과 온톨로지 주도 개발 방법론(Resource-Oriented and Ontology-Driven Development methodology)에 기반한 모델 주도 아키텍처(Model Driven Architecture)를 제안한다. 하지만 [16]은 기존의 SSNO에 액추에이터를 확장한 온톨로지를 제안할 뿐이므로 여전히 SSNO가 지니는 복잡한 구조와 애매모호함을 지닌다.

[17]은 IoT 환경에서 스마트 공간으로 센서와 액추에이터들을 분류하고 각 공간마다 이들을 관리하는 시스템을 제안한다. 하지만 공간별로 제한된 센서 정보만을 등록할 수 있어 다양한 의미와 센서들의 관계성을 충분히 고려하지 못한다.

3. SRS의 요구사항 정의

이 장은 SRS 데이터 모델을 설계하기 전에 SRS가 지니는 목적과 요구사항을 분석하고, 이를 기반으로 SRS의 유스케이스를 정의한다.

3.1 SRS의 목적 및 요구사항

기존의 SRS는 ISO/IEC 11179 메타데이터 레지스트리[18]의 개념을 적용하여 개발되었다. 이는 기본적으로 SRS가 센서 메타데이터를 등록하고 관리하는 것을 주된 목적으로 정의할 수 있다. 또한, 기존의 SRS는 이기종 센서 네트워크 환경을 고려한다. 이러한 환경에서는 하나의 센서 네트워크 영역에는 다수의 센서노드들이 존재하고, 이 센서노드들은 각자의 위치에서 싱크노드 혹은 사용자의 모바일 디바이스로 직접 데이터를 전달한다. 이때, 사용자가 다른 지역의 센

서 네트워크 영역으로 이동할 경우, 각 센서 네트워크 영역에서 발생하는 데이터의 이질성으로 인해 동일한 서비스를 제공받을 수 없다. 따라서 SRS는 센서노드의 메타데이터를 모바일 디바이스로 전달하여 각각의 영역에서도 의미처리가 가능해지도록 한다. 이처럼 기존의 SRS에서는 모바일 디바이스가 서로 다른 센서 네트워크 영역으로 이동하더라도 센서노드에서 보내는 센서 데이터에 대한 의미 처리가 끊김 없이(Seamless) 이루어지는 것을 목표로 한다.

하지만 IoT 패러다임이 부각되면서 센서 네트워크의 환경도 변화되었다. 모든 물체가 인터넷이 가능하게 되면서 기존의 센서 네트워크는 하나의 개체 단위로 소규모화 되고 그 종류도 다양해졌다. 이와 함께, SRS도 IoT의 환경에 맞추어 유스케이스와 요구사항 재정의가 필요하게 되었다. 다음은 IoT 환경에서 SRS가 요구하는 기능 및 목적이다.

- **다양한 센서와 이기종 네트워크 환경을 위한 정보 공유 및 유통.** 소규모화되고 다양해진 센서와 시스템들에 대하여 정보 공유와 유통이 원활하게 진행되어야 함.
- **센서의 접근성 향상.** 센서로부터 센서 데이터를 제공받고자 할 때, 플랫폼, 통신 방식, 센서의 위치 등 정보를 미리 알고 있어야 함.
- **센서 데이터의 끊김 없는 의미 해석.** 기존 SRS가 지니는 특징으로 모바일 디바이스가 어떠한 센서 데이터를 제공받더라도 의미 해석 및 처리가 끊김 없이 진행되어야 함.
- **센서 메타데이터 등록과 관리.** 단순히 센서 메타데이터를 등록하고 공유하는 것뿐만 아니라 고품질의 정보 제공을 위하여 등록된 센서 메타데이터들을 정제하고 관리되어야 함.

첫 번째 목적을 위하여 SRS는 센서 메타데이터 등록을 위한 플랫폼 및 어플리케이션 형태로 개발되어야 한다. 특히 모바일 디바이스가 어디서나 SRS에 접속이 가능하도록 센서 웹과 같이 웹 어플리케이션으로 개발한다. 이때, 등록된 센서 메타데이터의 일관성 유지와 빠른 성능의 정보 검색을 위하여 온톨로지 저장소 형태보다는 RDB로 데이터 모델을 구축한다. 하지만 센서 온톨로지로의 확장 및 추론을 고려하여 URI와 같은 정보도 등록 가능하도록 개발한다.

두 번째와 세 번째 목적을 위하여 기존에 잘 정의된 센서 모델을 활용하여 SRS의 데이터 모델을 개선한다. 센서의 접근성 향상을 위하여 시스템 및 센서 플랫폼이 잘 표현되어야 하며 센서 데이터에 대한 의미 정보를 충분히 담고 있어야 한다. SSNO는 이러한 내용이 잘 포함되어 있으나 복잡한 구조로 인해 RDB 테이블로 구축하기 어렵다. 따라서 SRS의 목적에 적합하도록 SSNO를 반영해야 한다. 이는 4장에서 자세하게 기술한다.

네 번째 목적을 위하여 각 센서 메타데이터들을 아이템화하고 관리 정보를 부여해야 한다. 예를 들면, 아이템이 등록된 시간, 아이템 등록자 및 관리자 등과 같은 관리 정보가

포함되어야 한다. 덧붙여, 로그 및 프로비넌스와 같은 정보들은 관리하는 아이템들의 신뢰성을 확보하기 위하여 함께 모델링 한다.

3.2 SRS의 유스케이스

앞 절에서 기술한 요구사항으로부터 SRS가 가져야 할 기능 및 이해당사자(Stakeholders)는 Fig. 1과 같이 표현할 수 있다.

먼저 제출자(Submitter)는 센서의 위치와 종류에 따라서 SRS로 센서 메타데이터를 등록하기 위한 정보들을 제출한다. 제출자는 센서 정보를 제출(Sensor Submission)하고, 제출한 정보를 관리(Sensor Management)하며, 기존의 정보들을 열람(Information View)할 수 있으며, 센서의 위치에 따라 지도상에서 열람(Map View)도 가능하다.

관리자(Steward)는 SRS에 관리되고 있는 아이템들을 관리한다. 관리자는 제출자로부터 SRS로 제출된 센서 정보들을 승인하여 등록(Sensor Registration)하며, 제출자와 마찬가지로 센서 관리, 정보 열람, 지도 열람 등이 가능하다.

마지막으로 사용자(User)는 기존의 등록되고 접근 가능한 센서 정보들을 열람하고 이용한다. 사용자는 제출자나 관리자와 같이 센서 정보를 제출하거나 관리하지는 못하지만 등록되어 있는 아이템의 정보를 열람할 수 있다. 또한 센서로 직접 접근하기 위하여 접근 정보를 요청하여 받는 서비스(Access Information Service)나 의미 정보를 요청하여 받는 서비스(Semantic Information Service)를 이용할 수 있다.

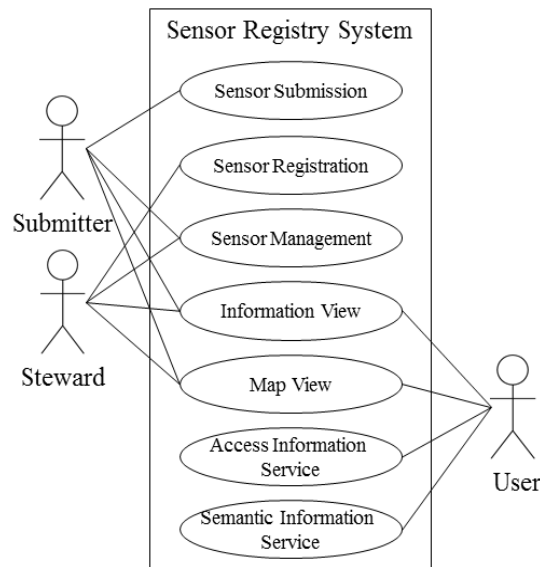


Fig. 1. Use case of SRS

추가적으로, SRS는 IoT 환경에 맞추어 웹 어플리케이션으로 개발하는 것이 요구되며, Linked Open Data (LOD)와 같은 의미적 확장을 위하여 URI도 이용할 수 있도록 한다. 또한, Restful API 등을 이용한 UI를 탑재함으로써 실시간으로 동적인 센서 정보를 획득 가능하게 한다.

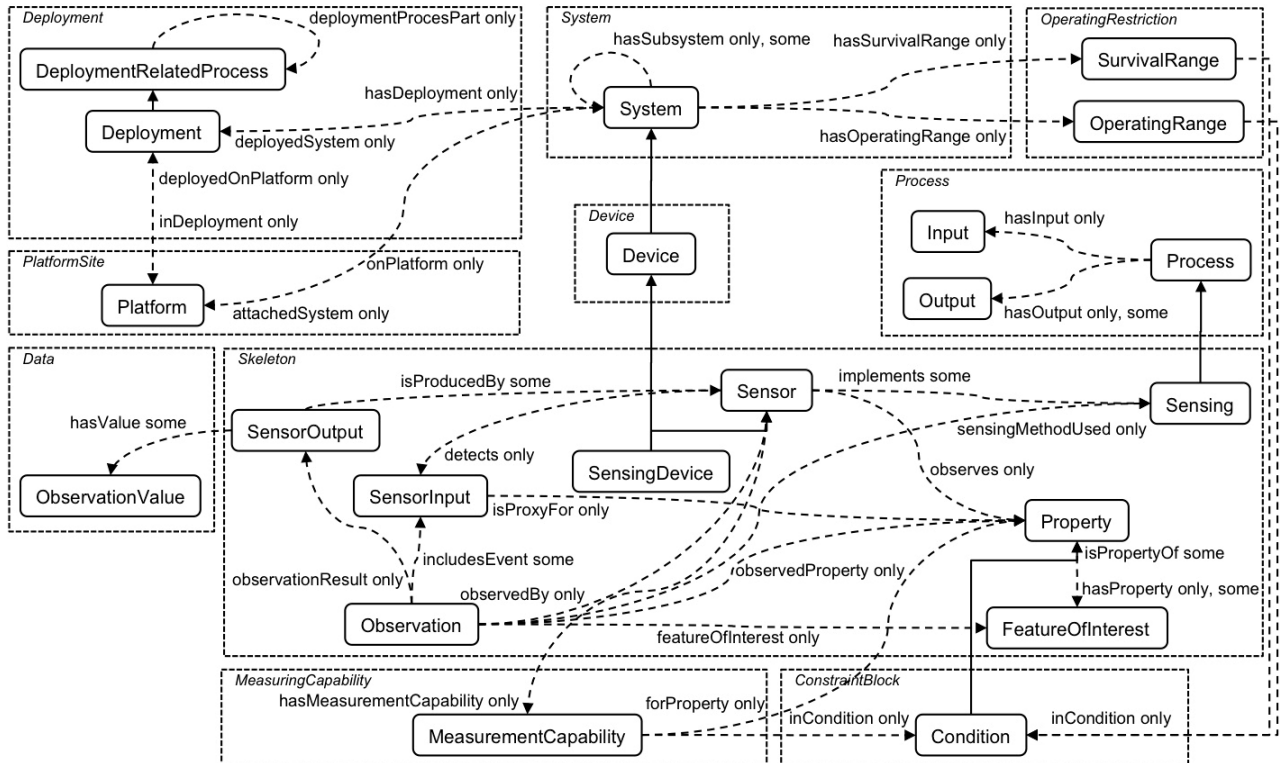


Fig. 2. Overview of SSNO – main classes and attributes [10]

4. 센서 레지스트리 데이터 모델링

이 장은 SRS의 데이터 모델을 설계하기 위하여 SRS의 목적에 맞추어 SSNO를 분석한다. 분석 결과를 이용하여 메타모델을 설계하고, 이를 다시 RDB 테이블로 생성한다.

4.1 메타모델 설계

SSNO는 W3C에서 표준화를 위해 약 17여 종류의 센서 온톨로지들을 분석하고, 일반적이고 포괄적인 관점과 의미를 포함하도록 모델링 된 온톨로지이다. 하지만 SRS에 적용하기엔 너무 일반적이고 복잡한 구조를 지니므로 SSNO를 이용하여 SRS에 적합한 데이터 모델을 설계 하고자 한다.

Fig. 2는 W3C에서 정의한 SSNO의 개요로 주요 클래스들과 속성들을 보인다[10]. 먼저 두 번째 목적인 센서 접근성 향상을 위해 SSNO에서 이용할 수 있는 요소들을 정의한다. 실제 임의의 위치에 설치된 시스템 인스턴스를 나타내는 ‘System’과 플랫폼 정보를 나타내는 ‘Platform’은 SR 데이터 모델에 센서 접근성을 위하여 필요한 요소들이다. ‘Deployment’는 시스템이 설치되는 과정, 방법 등과 연관이 있지만, 센서의 접근성과는 관계가 없으므로 제외한다. 하지만 설치된 시스템을 관리하는 업체와 시스템이 설치된 위치 SSNO 개요에 직접적으로 표현되지 않으므로 각각 새로 정의한다.

다음으로 세 번째 목적인 센서 데이터의 끊임 없는 의미 해석을 위한 요소들을 정의한다. ‘Sensor’는 어떤 특징과 속

성을 관찰하는 요소로 추후 제공받을 센서 데이터의 주체가 된다. ‘Property’의 경우 어떠한 특징 및 속성을 관찰하는지를 나타내는 요소이다. 이 두 가지 요소는 센서가 무엇을 관측하며 그 값이 어떤 의미를 지니는지를 해석하는데 필수적인 요소이므로 SR 데이터 모델에 필요하다.

‘Device’, ‘SensingDevice’의 경우 중복된 정의와 다중 상속으로 인해 복잡한 구조를 보인다. 이는 RDB 테이블 구축시 중복되는 구조가 발생할 수 있어 제외하고 ‘Sensor’를 ‘System’ 요소와 직접 연결되도록 한다. 또한 SRS는 센서 메타데이터를 제공할 뿐 실제 값이나 값이 수집되는 프로세스 등과는 연관이 없으므로 ‘Observation’이나 ‘Process’와 연관된 요소들은 모두 제외한다. 센서 데이터가 지니는 단위와 데이터 타입에 관련된 요소는 의미 해석에 필수적이므로 추가한다.

분석된 결과를 이용하여 클래스 다이어그램 기반의 메타모델을 정의한다. 메타모델은 클래스 다이어그램에 기반하기 때문에 기존의 온톨로지 보다는 다양한 표현이 가능하다. 따라서 이러한 메타모델은 기존의 SensorML[11]이나 ISO/IEC 11179 메타데이터 레지스트리와 같은 표준에서 자주 이용된다.

Fig. 3은 앞서 기술한 SSNO 분석 결과를 이용하여 SRS에 이용할 개념들과 제외할 개념들을 메타모델로 모델링 한 결과이다. ‘SRSTypes Package’는 SSNO에 기반한 센서 메타데이터를 정의한 패키지, 이 패키지 안의 클래스 및 관계들은 센서의 접속 정보 및 의미 정보를 포함한다.

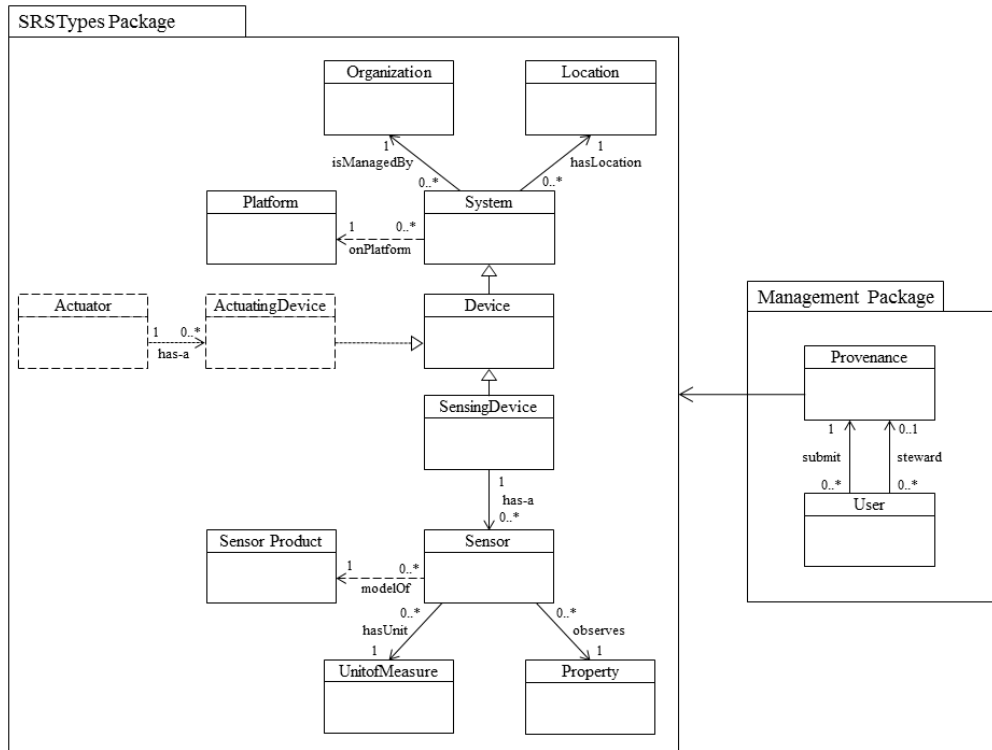


Fig. 3. Metamodel for SRS

‘System’, ‘Device’, ‘Platform’, ‘Sensor’, ‘Property’와 같은 클래스들은 SSNO의 개념 그대로 이용하고, 위치를 나타내는 ‘Location’과 센서 및 시스템을 관리하기 위한 ‘Organization’을 추가한다. 또한 ‘Sensor’가 측정하는 값의 단위인 ‘UnitofMeasure’를 추가한다. ‘SensorProduct’는 제품으로써의 센서 정보이다. 대부분의 센서들은 완제품으로 시스템에 장착되기 때문에 시스템마다 동일한 센서 정보를 따로 입력하는 번거로움을 줄이기 위해 추가되었다. ‘Device’의 경우, 추후 ‘ActuatingDevice’ 및 ‘Actuator’와 같이 액추에이터 등록을 위한 확장 가능성을 열어두기 위하여 정의하였으며, 이러한 특성은 ‘SensingDevice’와의 차이점을 명확히 한다. 또한, SSNO에서의 ‘SensingDevice’는 ‘Device’와 ‘Sensor’를 이중상속 하지만, 이러한 애매모호한 표현을 피하기 위하여 하나의 ‘SensingDevice’가 여러 개의 ‘Sensor’를 이용할 수 있도록 연관관계로 정의한다.

‘Management Package’는 ‘SRSTypes Package’에서 관리되는 모든 정보들을 관리하기 위한 프로비던스 정보 및 사용자 정보 등을 정의한 패키지이다. ‘Provenance’와 ‘User’ 클래스는 구현될 SRS 어플리케이션에서 각 센서 정보들이 등록 및 관리될 때 추가적인 정보들을 담는다.

4.2 관계형 데이터베이스 모델

SRS의 구현을 위해서 제안 데이터 모델을 RDB 테이블로 구축한다. SRS 메타모델을 RDB 기반의 데이터 모델로 구현하는 이유는 3.1절의 첫 번째 요구사항 분석에서 기술하였다.

요소 별 기본 속성을 정의하기 위하여 먼저 각 요소들의 인스턴스를 하나의 아이টে姆으로 정의한다. 따라서 각 요소들은 아이টে姆 식별자(id), 이름(name), 설명(description)과 같은 속성들을 기본적으로 지닌다. 또한 온톨로지 확장을 위한 URI(uri)도 기본 속성으로 정의한다. 이러한 아이টে姆들은 등록 및 관리 시 얻어질 수 있는 부수적인 정보들을 프로비던스 요소로서 저장한다. 추가적으로 사용자에 대한 요소도 모델링 한다.

Fig. 4는 이러한 기준으로 모델링한 SRS 데이터 모델을 보인다. RDB의 테이블, 컬럼, 외래 키(FK) 관계로 각 요소들을 모델링한다. 먼저 ‘System’ 테이블을 기준으로 ‘Platform’, ‘Organization’, ‘Location’ 등과 같은 테이블을 정의한다. 이 세 테이블들은 ‘System’ 테이블이 지니는 속성으로 외래 키 관계로 구성된다. ‘Sensor’의 경우 기존의 이중 상속의 복잡한 구조를 벗어나 하나의 시스템이 여러 개의 센서를 지니도록 구성한다. 예를 들어, TelosB MOTE TPR2420CA의 경우 하나의 시스템에 조도, 습도, 온도와 같은 세 가지 센서를 지니고 있다. 따라서 ‘Sensor’ 테이블은 ‘System’ 테이블을 외래 키 관계로 연결한다. 또한 측정하는 대상의 특징 및 데이터의 의미 처리를 위하여 ‘UnitofMeasure’와 ‘Property’ 테이블을 외래 키 관계로 연결한다. 한편, ‘Sensor_Product’ 테이블은 단품으로 나온 센서 제품을 등록하기 위하여 제품의 모델명(name), 생산 회사(organization) 등을 속성으로 추가한다. ‘Provenance’와 ‘User’ 테이블은 등록되는 아이টে姆에 대한 로그 및 프로비던스 정보와 함께 사용자에 대한 정보를 저장한다.

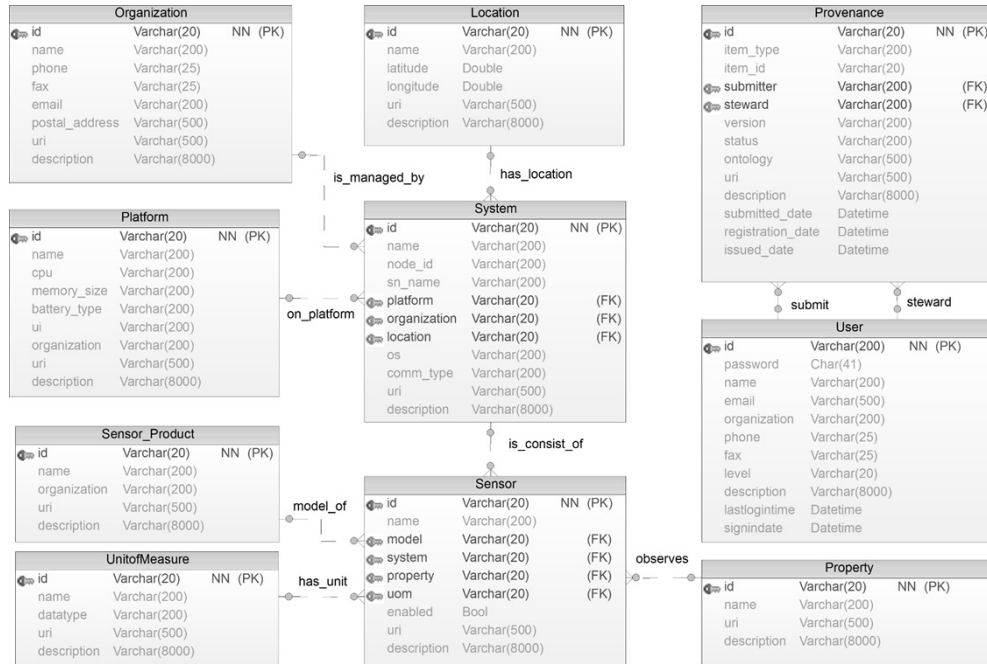


Fig. 4. Proposed SRS data model

5. 센서 레지스트리 시스템 구현

앞서 설계한 SR 데이터 모델을 이용하여 실제 RDB 테이블을 구축하고 웹 어플리케이션으로 SRS를 구현한다. Windows 7, Mysql 5.5, Apache Tomcat 7.0 환경에서 시스템을 구축하였으며 JSP와 HTML5를 이용하여 개발하였다.

Fig. 5~9는 SRS의 구현 결과를 보인다. Fig. 5는 센서 레지스트리 시스템의 전체 화면과 등록된 시스템들의 간단한 리스트를 보인다. Fig. 6은 등록된 센서 중 하나에 대한 자세한 정보이며 'Model', 'System', 'Property'와 같이 해당 센서가 지니는 다른 테이블로 연결된 정보까지 요약해서 보여준다. Fig. 7은 Fig. 6의 아이템 관리 정보를 보인다. 기존의 로그 이력과 등록자, 관리자 등을 알 수 있다. Fig. 8은 센서 맵을 구현한 것으로 시스템 및 센서가 장착된 장소 지도를 통하여 보인다. 마지막으로 Fig. 9는 지도상에 위치한 센서 마크를 클릭했을 때 해당 위치에서의 시스템 및 센서 정보를 자세히 보인다.

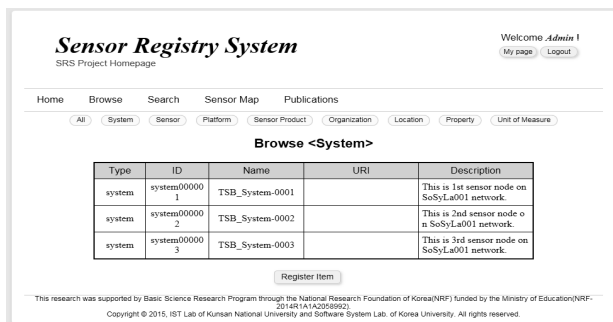


Fig. 5. A screenshot of system list

Sensor <Light_S1087>

ID	sensor000001
Name	Light_S1087
Model	ID: sproc000002
	Name: S1087
	Organization: Hamamatsu
	URI: http://www.hamamatsu.com/en/product/category:310040014103/S1087/index.html
System	ID: system000001
	Name: TSB_System-0001
	Node ID: 1
	Sensor network name: SoSyLa001
	Platform: platform000001
	Management Organization: organ000001
	Location: location000001
	OS: TinyOS
Property	ID: property000005
	Name: Brightness
	URI:
	Description: measuring illumination by light sensor.
Unit of Measure	
Enabled	true
URI	
Description	light system

Fig. 6. A screenshot of sensor item information view

Item Information	
Item Type	sensor
Item ID	sensor000001
Submitter	a
Steward	
Version	v000002
Status	submitted
Ontology	
Submitted Date	2015-07-03 01:17:42.0
Registered Date	
Issued Date	2015-07-13 16:48:59.0
URI	
Description	

Fig. 7. A screenshot of the provenance and log view

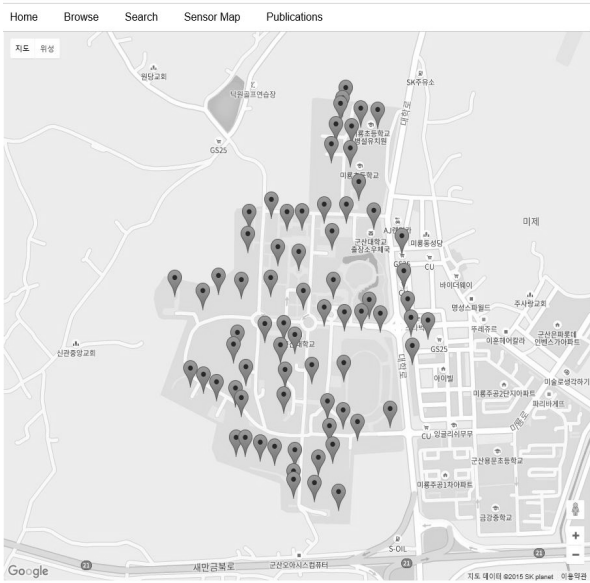


Fig. 8. A screenshot of sensor locations on the map

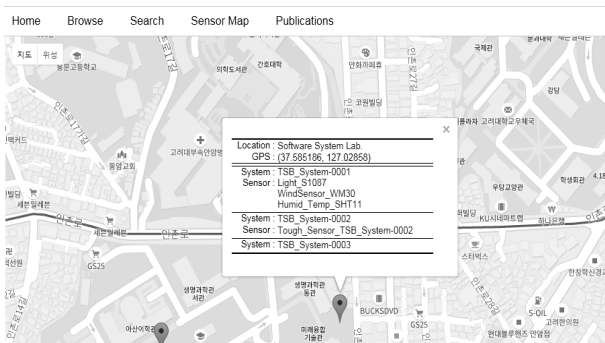


Fig. 9. A screenshot of sensor information view on the map

6. 비교 평가

이 장은 제안하는 SR 데이터 모델의 적합성을 검증하고 다른 센서 플랫폼들과 비교를 통하여 제안 시스템의 정성 평가를 기술한다.

6.1 적합성 평가

적합성 평가는 제안하는 SR 데이터 모델과 기존 센서 데이터 모델들의 센서 온톨로지 이식률 측정으로 이루어진다. 온톨로지 이식을 위하여 SSNO와 몇 가지 센서 온톨로지 예제들을 기존 연구들과 제안 모델로 변환시킨다. 이때, 이식률은 온톨로지가 담고 있는 의미정보들을 각 모델들이 얼마나 표현할 수 있는지로 측정된다. 즉, SSNO 및 센서 온톨로지 예제들이 지니는 온톨로지의 개념들(클래스, 속성, 인스턴스)을 각 모델들로 변환시켰을 때 변환이 가능한 개념들의 개수로 측정한다.

비교 평가를 위하여 관련 연구들 중 RDB를 이용하고, 시

맨틱 센서 네트워크를 대상으로 하는 연구를 선정한다. 따라서 비교 대상으로 기존 SRS의 데이터 모델(Jeong *et al.*)[9]과 스마트 공간에서의 센서 모델(Choi *et al.*)[17]을 선정하였다. 평가 모델로는 SSNO의 주요 클래스 및 속성들과 에너지 모듈, TelosB 센서노드, 바람 측정 센서에 대한 온톨로지 예제들을 선정하였다[13].

Table 1은 온톨로지 변환 개수로 비교 평가 결과를 보인다. 온톨로지들은 총 157개의 클래스, 속성, 인스턴스 등을 지니며, Jeong, Choi, 제안 모델은 각각 총 47, 46, 62개의 요소가 변환되었다. 제안 모델은 가장 많은 개념들이 변환되어 다른 모델들에 비해 시스템과 플랫폼을 구체적으로 표현하였음을 의미한다. 특히 Choi의 모델은 센서로부터 수집되는 값의 표현이 가능하나 각 요소 간 관계 표현이 불가능하다. 따라서 평가 결과 제안하는 SRS 데이터 모델은 다른 모델들 보다 더 풍부한 의미 표현이 가능함을 보인다.

Table 1. Results of a compatibility evaluation

Ontology	Jeong	Choi	Proposal
SSNO	7 / 49	10 / 49	12 / 49
Energy ontology	16 / 29	13 / 29	20 / 29
TSB ontology	11 / 34	15 / 34	14 / 34
WM30 ontology	13 / 45	8 / 45	16 / 45
Total	47 / 157	46 / 157	62 / 157

6.2 정성 평가

제안 데이터 모델의 우수성을 기술하기 위하여 구축한 시스템과 기존의 센서 플랫폼들의 기능과 특징을 비교한다. Table 2는 IoT 환경에서 고려되어야 할 센서 플랫폼들의 특성에 대한 지원 여부를 보인다. 비교 대상으로는 SensorMap[4], Sensor-Cloud[7], CA4IOT[8]을 선정하였다.

IoT 환경에서는 센서들이 소규모화된 시스템 안에서 다양한 종류의 센서가 함께 탑재된다. 센서의 개체 수 증가로 인하여 센서 검색이 용이하고 정보의 신뢰성이 보장되어야 한다. 따라서 정성 평가에서는 제안 시스템인 SRS와 선정된

Table 2. Supportable features of sensor platforms

Feature	SensorMap	Sensor-Cloud	CA4IOT	SRS (Proposed)
value observation	support	support	support	not support
sensor embedded system	not support	not support	not support	support
spatial position	support	not support	not support	support
actuator	not support	not support	not support	not support (considered)
provenance	not support	not support	not support	support
semantic process	not support	support	support	support

플랫폼들의 값 관측(value observation), 센서가 부착된 시스템(sensor embedded system), 지역적 위치(spatial position), 액추에이터(actuator), 출처 관리(provenance), 의미 처리(semantic process)에 대한 지원 여부를 측정하였다.

값 관측은 플랫폼이 센서로 부터 관측된 값을 제공해주는 지를 나타낸다. 다른 플랫폼들이 이러한 시스템을 지원해주는 반면, SRS는 센서의 메타데이터와 같은 의미정보만을 제공해주므로 값 관측은 지원하지 않는다. 다만 기존의 센서 플랫폼들이 주기적으로 직접 센서에 접근하여 관측 값을 얻어오던가, 센서가 주기적으로 관측 값을 푸시(push)하는 방식이라면, SRS는 센서로 직접 접근할 수 있는 정보를 제공하므로 사용자는 다른 플랫폼들을 통해 얻는 것보다 SRS를 이용하여 더 정확하고 빠르게 관측 값을 얻을 수 있다.

센서가 부착된 시스템은 하나의 시스템에 여러 센서의 표현을 지원하는지를 나타낸다. 기존의 센서 네트워크에서는 하나의 네트워크가 여러 센서노드들을 지니며 각 노드를 단순히 하나의 센서로만 인식하였다면, 이러한 표현 방식은 IoT 환경에서 하나의 디바이스 혹은 개체 자체를 시스템으로 인식하고, 이 시스템 안에 여러 센서가 내포될 수 있음을 의미한다. 다른 플랫폼들은 이러한 개념을 차용하지 않은 반면, SRS는 디바이스를 구성하는 시스템과 센서를 분리하여 표현함으로써 각 디바이스에 대한 접근성을 높인다.

지역적 위치는 센서가 이동하였을 때 이를 반영할 수 있는지를 나타낸다. SensorMap과 SRS는 센서 정보에 위도, 경도 좌표와 같이 지역적 위치를 지니고 있어 센서가 이동하였을 때 이를 수정하고 표시할 수 있다. 하지만 Sensor-Cloud와 CA4IOT는 어떤 센서 네트워크에 포함되는지가 중요하며 지역적 위치는 중요한 요소로 작용하지 않는다.

액추에이터는 센서의 관측된 값이나 관측에 필요한 정보뿐만 아니라 관측 후 어떤 행동을 수행할지 결정할 수 있다. 즉, 하나의 디바이스가 센서뿐만아니라 액추에이터도 함께 지니는지를 표현하고 지원하는지를 나타낸다. 모든 플랫폼들이 이를 지원하지 않고 있지만, SRS는 Fig. 3을 통하여 액추에이터를 추후에 추가할 사항이라는 것을 염두해 두고 있다.

출처 관리는 센서를 등록하고 정보를 수정할 때 믿을만한 지를 나타낸다. 다른 플랫폼들은 정보 변경의 이력이나 관리자 및 관측자에 대한 접근제어 정도까지 표현하지만, SRS는 입력한 정보의 정확한 출처 정보도 입력 가능하게 개발되었다.

마지막으로 의미 처리는 센서 정보들을 이용하여 센서의 의미를 파악하고 처리하는 것이 고려되었는지를 나타낸다. SensorMap이 단순히 위치에 따른 센서들을 표시하고 해당 센서에 대한 값과 메타데이터를 제공해주는 정도였다면, 다른 플랫폼들은 온톨로지 등을 이용하여 의미정보를 처리할 수 있도록 한다. 특히 CA4IOT와 같은 경우는 센서 검색, 문맥 저장소, 추론기 등을 통하여 사용자에게 필요한 의미 있는 센서 정보를 제공하여 분석에 이용한다.

이외에도, 사용자의 서비스 반응 속도, 플랫폼 이용 및 센서 등록의 용이성 등과 같이 플랫폼 간 고려해야 할 특성들이 다수 존재한다. 하지만 IoT 환경에서 지원해야 할 특성을 고려해 보았을 때 제안하는 SRS가 기존의 센서 플랫폼들보다 더 적합함을 보인다.

7. 결 론

이 논문은 기존의 SRS가 IoT 환경을 충분히 적용하지 못하는 문제를 해결하기 위하여 IoT 환경에 적용 가능한 SR 데이터 모델을 설계하였다. 먼저 SRS의 목적 및 요구사항을 분석하고, 이를 기반으로 구현하고자 하는 SRS의 유스케이스를 기술하였다. SR 데이터 모델링을 위하여 SSNO를 이용하여 메타모델을 설계하였으며, 구현을 위하여 RDB 테이블로 구축하였다. 이후 SRS를 웹 어플리케이션으로 구현하였다. 평가를 위하여 기존 센서 모델들과 제안 모델의 적합성을 평가하였으며, SSNO와 3가지 온톨로지 예제들을 변형시켜 각 모델에 적용해 본 결과 제안 모델이 가장 높은 적합성을 보임을 검증하였다. 정성 평가로는 기존의 센서 플랫폼들과 SRS를 대상으로 IoT 환경에서 지원해야 할 기능과 특성들을 비교하였다.

추후 연구로는 모바일 디바이스, SRS, 센서가 연계하여 사용자가 실시간으로 제공받을 수 있는 서비스를 개발하고 정량 평가를 통한 시스템 검증이 요구된다.

References

- [1] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey," *Computer Networks*, Vol.54, pp.2787-2805, 2010.
- [2] P. Pande and A. R. Padwalkar, "Internet of Things - A Future of Internet: A Survey," *International Journal of Advance Research in Computer Science and Management Studies*, Vol.2, No.2, pp.354-361, 2014.
- [3] C. Reed, M. Botts, G. Percivall, and J. Davidson, "OGC Sensor Web Enablement: Overview And High Level Architecture," *OGC White Paper, Open Geospatial Consortium*, Apr., 2013.
- [4] S. Nath, J. Liu, and F. Zhao, "SensorMap for Wide-Area Sensor Webs," *IEEE Computer*, Vol.40, No.7, pp.106-109, Jul., 2007.
- [5] B. L. Gorman, D. R. Resseguie, and C. Tomkins-Tinch, "Sensorpedia: Information Sharing Across Incompatible Sensor Systems," in *Proceedings of International Symposium on Collaborative Technologies and Systems 2009*, Baltimore, Maryland, USA, pp.448-454, May, 2009.
- [6] X. Sheng, J. Tang, X. Xiao, and G. Xue, "Sensing as a Service: Challenges, Solutions and Future Directions," *IEEE Sensors Journal*, Vol.13, No.10, pp.3733-3741, Oct., 2013.

[7] A. Alamri, W. S. Ansari, M. M. Hassan, M. S. Hossain, A. Alelaiwi, and M. A. Hossain, "A Survey on Sensor-Cloud: Architecture, Applications, and Approaches," *International Journal of Distributed Sensor Networks*, Vol.2013, article ID 917923, pp.1-18, 2013.

[8] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos, "CA4IOT: Context Awareness for Internet of Things," in *Proceedings of 2012 IEEE International Conference on Green Computing and Communications, Conference on Internet of Things, and Conference on Cyber, Physical and Social Computing*, pp.775-782, Nov., 2012.

[9] D. Jeong and J. Ji, "A Registration and Management System for Consistently Interpreting Semantics of Sensor Information in Heterogeneous Sensor Network Environments," *Journal of KIISE: Databases*, Vol.38, No.5, pp.289-302, 2011. (in Korean).

[10] M. Compton et al., "The SSN ontology of the W3C semantic sensor network incubator group," *Web Semantics: Science, Services and Agents on the World Wide Web*, Vol.17, pp. 25-32, 2012.

[11] Open Geospatial Consortium, OpenGIS Sensor Model Language (SensorML) Implementation Specification, OGC 07-000, OGC, 2007.

[12] M. Compton, C. Henson, L. Lefort, H. Neuhaus, and A. Sheth, "A survey of the semantic specification of sensors," in *Proceedings of 2nd International Semantic Sensor Networks Workshop, International Workshop on Semantic Sensor Networks 2009*, Washington DC, USA, pp.17-32, Oct., 26, 2009.

[13] W3C Incubator Group, Semantic Sensor Network XG Final Report [Internet], <http://www.w3.org/2005/Incubator/ssn/XGR-ssn-20110628/>.

[14] European Lighthouse Integrated Project, Internet of Things - Architecture (IoT-A) [Internet], <http://www.iot-a.eu/public>.

[15] Semance AG, Internet of Things - Architecture Project Deliverable D1.2 - Initial Architectural Reference Model for IoT, IoT-A 257521, European Lighthouse Integrated Project, 2011.

[16] I. Corredor, A. M. Bernardos, J. Iglesias, and J. R. Casar, "Model-Driven Methodology for Rapid Deployment of Smart Spaces Based on Resource-Oriented Architectures," *Sensors* 2012, Vol.12, No.7, pp.9286-9335, 2012.

[17] H.-S. Choi and W.-S. Rhee, "IoT-Based User-Driven Service Modeling Environment for a Smart Space Management System," *Sensors*, Vol.14, No.11, pp.22039-22064, 2014.

[18] ISO/IEC JTC 1/SC 32, ISO/IEC 11179-3:2013 - Information Technology - Metadata Registries (MDR) - Part 3: Registry Metamodel and Basic Attributes, 3rd ed., 2013.



이 석 훈

e-mail : leha82@ajou.ac.kr

2009년 고려대학교 전자및정보공학부 (학사)

2011년 고려대학교 컴퓨터·전파통신공학과 (석사)

2016년 고려대학교 컴퓨터·전파통신공학과 (박사)

2016년~현 재 아주대학교 의료정보학과 연구강사(Post-Doc.)
 관심분야: 온톨로지, 센서 네트워크, 사물인터넷, 의료 빅데이터, 생체신호 데이터



정 동 원

e-mail : djeong@kunsan.ac.kr

1997년 군산대학교 컴퓨터학과(학사)

1999년 충북대학교 전산학과(석사)

2004년 고려대학교 컴퓨터학과(박사)

2005년~현 재 군산대학교 통계컴퓨터
 과학과 교수

관심분야: 데이터베이스, 시맨틱 서비스, 빅데이터, 사물인터넷



정 현 준

e-mail : darkspen@korea.ac.kr

2008년 삼육대학교 컴퓨터학과(학사)

2010년 숭실대학교 컴퓨터학과(석사)

2011년~현 재 고려대학교 컴퓨터·전파
 통신공학과 박사과정

관심분야: 데이터베이스, 빅데이터, 사물
 인터넷



백 두 권

e-mail : baikdk@korea.ac.kr

1974년 고려대학교 수학과(학사)

1977년 고려대학교 산업공학과(석사)

1983년 Wayne State Univ. 전산학과(석사)

1985년 Wayne State Univ. 전산학과(박사)

현 재 고려대학교 컴퓨터학과 교수

1989년~1991년 고려대학교 전산학과 학과장
 1990년~1991년 미국 Arizona대학교 객원 교수
 1991년~2013년 ISO/IEC JTC1/SC32 전문위원회 위원장
 1993년~1999년 한국과학기술원 객원책임연구원
 1993년~1999년 한국DB진흥센터 표준연구위원
 1996년~1997년 고려대학교 컴퓨터과학기술연구소(초대소장)

1997년~1998년 고려대학교 정보전산원 원장
1998년~1999년 한국정보과학회 전산교육연구회 운영위원장
1999년~2001년 정보통신진흥협회 데이터기술위원회 의장
2002년~2004년 고려대학교 정보통신대학(초대학장)
2002년~2003년 한국시뮬레이션학회 회장
2003년~현 재 정보통신부 컴퓨터프로그램보호위원회 위원
2004년~2005년 한국정보처리학회 부회장
2005년~2008년 한국소프트웨어진흥원 이사
2009년~2010년 고려대학교 정보통신대학 학장
관심분야: 메타데이터, 소프트웨어공학, 데이터공학, 컴포넌트기반
시스템, 메타데이터 레지스트리, 프로젝트 매니지먼트