

# 센서 메타데이터 영역화 및 재사용성 기반 센서 레지스트리 시스템 성능 향상 방법

정동원<sup>†</sup>

## 요 약

센서 레지스트리 시스템은 이기종 센서 네트워크에 독립적으로 센서 데이터 의미를 해석하고 처리하기 위해 제안되었다. 그러나 기존 센서 레지스트리 시스템 구조는 정적 처리 방법을 제공한다. 즉, 이용할 데이터 영역을 고려하지 않고 불필요한 연산을 수행함으로써 전체적인 처리 성능을 저하시킨다. 이 논문에서는 기존 센서 레지스트리 시스템 구조의 문제점을 해결하기 위해 센서 메타데이터 영역화와 재사용성에 기반한 성능 개선 모델을 제안한다. 제안 모델은 센서 레지스트리 시스템에서 모바일 디바이스로 전송되는 센서 메타데이터의 영역을 상황 맞게 결정할 수 있는 기능을 제공한다. 또한 센서 메타데이터의 재사용성을 지원함으로써 전체적인 성능을 향상시킨다. 마지막으로 이 논문에서는 성능 비교 평가를 통해 제안 모델의 장점을 보인다.

주제어 : 유비쿼터스 컴퓨팅, 센서 네트워크, 레지스트리, 메타데이터, 성능

## Performance Improvement of the Sensor Registry System based on Sensor Metadata Reusability and Scoping

Dongwon Jeong<sup>†</sup>

### ABSTRACT

The sensor registry system has been proposed to interpret and process semantics of sensor data independently of heterogeneous sensor networks. However, the existing sensor registry system provides the static processing method. In other words, the existing system reduces the overall performance because it executes unnecessary operations and does not consider data scope to be used. To resolve the problem of the existing sensor registry system, this paper proposes a performance enhancement model based on sensor metadata reusability and scoping. The proposed model in this paper provides a function that can decide a proper scope of sensor metadata from the sensor registry system. The proposed model improves the overall performance by providing reusability of sensor metadata. This paper also shows the advantages of the proposed model through the comparative performance evaluation..

**Keywords** : Ubiquitous Computing, Sensor Network, Registry, Metadata, Performance

---

<sup>†</sup> 중신회원: 군산대학교(교신저자)

논문접수: 2012년 10월 18일, 심사완료: 2012년 10월 30일, 게재확정: 2012년 11월 01일

\* 본 논문은 2012년도 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임.  
(NO.2012-0002439)

## 1. 서론

유비쿼터스 센서 네트워크 환경은 시간, 장소 및 디바이스에 무관하게 센서 정보를 이용할 수 있어야 한다. 특히 이기종 센서 네트워크에 독립적으로 센서 데이터의 의미를 해석하고 처리할 수 있어야 한다. 현재까지 이기종 센서 네트워크 상에 분포되어 있는 동질, 이종의 센서 데이터를 이용하기 위한 다양한 연구들이 진행되어 왔다 [1][2][3][4][5][6]. 그러나 이러한 연구들은 사전에 의미 처리가 이루어진 상태에서 센서 정보를 이용하는 구조이다. 즉, 모바일 디바이스에서 직접 센서 정보를 해석하지 못한다는 문제점과 함께 이질적인 센서들로부터 수신된 정보 의미를 모바일 디바이스에서 직접 해석할 수 없다는 문제점을 지닌다.

이러한 문제점을 해결하기 위해 [7]에서는 센서 레지스트리 시스템(Sensor Registry System, SRS)을 제안하고 있다. 센서 레지스트리 시스템은 모바일 디바이스에서 센서 데이터 및 다양한 센서 관련 정보(센서 메타데이터)를 처리할 수 있는 기능을 제공하며, 보다 진화된 유비쿼터스 컴퓨팅 환경 구현을 가능하게 한다.

센서 레지스트리 시스템의 장점에도 불구하고, [7]에서 제안한 센서 레지스트리 시스템 아키텍처는 전체적인 성능, 즉 처리 시간이 저하되는 문제점을 지닌다. 또한 센서 웹과 같은 서비스에서 제공하는 센서 정보의 활용성을 고려하지 않고 있다.

[8]에서는 여러 가지 요인을 고려한 센서 레지스트리 시스템 구성 방안을 제시하고 있다. 그러나 이 논문에서는 고려해야 할 요인 및 요인에 따른 시스템 구성 방법에 대한 개념만을 제시하고 있을 뿐, 요인에 대한 타당성 및 구체적인 처리 연산 구조를 제시하지 않고 있다. 무엇보다 기존 센서 레지스트리 시스템과의 정량적 성능 평가 결과를 제공하지 않는다.

따라서 이 논문에서는 [8]에서 정의한 요인 중에서, 성능에 직접적인 영향을 주는 센서 메타데이터 영역을 고려한 구체적인 센서 레지스트리 시스템의 성능 개선 방법을 기술한다. 이 논문에서 제안하는 성능 향상 방법의 핵심은 센서 메타

데이터의 재사용성과 영역화에 기반한 보다 적응적인 센서 메타데이터의 활용이다. 이 논문에서는 기존 센서 레지스트리 시스템의 성능을 향상시킬 수 있는 모델을 기술하고 비교 평가를 통한 제안 방법의 장점을 보인다.

이 논문의 구성은 다음과 같다. 제2장에서는 [7]에서 제안한 센서 레지스트리 시스템의 문제점을 상세하게 서술하고 제3장에서는 주요 용어를 정의하고 성능 개선 방법의 개념, 제안 모델의 특징 및 주요 알고리즘에 대하여 기술한다. 제4장에서는 성능 평가 모델 및 비교 평가 결과 대하여 기술한다. 마지막으로, 제5장에서는 결론 및 향후 연구에 대하여 서술한다.

## 2. 문제 정의

이 장에서는 [7]에서 정의한 센서 레지스트리 프레임워크의 전체적인 동작 과정 및 문제점을 상세하게 서술한다.

센서 레지스트리 시스템에서, 디바이스는 센서로부터 센서 데이터를 수신하며, 센서의 유형, 값의 처리 단위 등에 대한 해석을 위해 필요한 정보(메타데이터)를 센서 레지스트리 시스템에 요청한다. 센서 레지스트리 시스템은 모바일 디바이스에서 요청한 해당 센서의 메타데이터를 검색하여 결과를 디바이스에 전송한다. 디바이스는 전송된 정보를 이용하여 센서 데이터를 해석하여 원하는 서비스 결과 제공에 이용한다.

다음은 단일 센서로부터 수신된 센서 데이터 이용과정을 의사코드로 표현한 알고리즘이며, 이 알고리즘을 통해 기존 센서 레지스트리 시스템의 문제점을 상세하게 분석한다.

```
START_FUNCTION (INPUT s1: Sensor, srs: SRS)
  sData: SensorData;
  sMeta: SensorMetadata;
  loop: Boolean; tag←TRUE;
START_WHILE(loop)
  sData←s1.receiveSensorData();
  sMeta←srs.getMetadata(sData.getID());
  interpretSensorData(sData, sMeta);
  loop←utilizeSensorInfo(sData, sMeta);
```

```
END_WHILE
END_FUNCTION
```

위 알고리즘에서, 먼저 모바일 디바이스는 receiveSensorData() 연산을 통해 센서로부터 센서 데이터를 수신한다. 수신한 센서 데이터(sData)는 센서가 감지한 센서 값과 센서 ID의 쌍(SID, value)을 구성된다. 모바일 디바이스는 센서로부터 획득한 센서 값을 해석하기 위해 센서 ID를 이용하여 센서 레지스트리 시스템에 해당 센서의 설명 정보에 해당하는 센서 메타데이터를 요청한다. 이 때, 센서 레지스트리 시스템은 센서 ID와 동일한 관련 모든 센서 메타데이터(sMeta)를 모바일 디바이스에 전송한다. 전송된 센서 메타데이터를 이용하여 센서 유형 및 값의 단위를 해석하여 활용 연산에 이용한다.

위 알고리즘에서 보여준 기존 센서 레지스트리 시스템의 처리 과정은 크게 두 가지 문제점을 야기한다.

첫 번째는 모바일 디바이스에 모든 센서 메타데이터를 전송한다는 점이다. [7]에 정의되어 있듯이, 센서 레지스트리 시스템의 센서 메타데이터는 기본 센서 메타데이터와 응용 센서 메타데이터로 구성된다. 기본 센서 메타데이터는 센서의 유형, 센서 값의 단위 등과 같은 센서 데이터를 해석하기 위해 가장 기본적으로 필요한 정보이며, 응용 센서 메타데이터는 센서 제조사, 설치 기관, 관리 기관 등 센서와 관련된 추가적인 정보를 포함한다. 따라서 모바일 디바이스에서 센서 데이터 해석에 필요하지 않은 정보를 전송함으로써 모바일 디바이스는 물론 센서 레지스트리 시스템의 성능을 저하시킨다.

두 번째 문제점으로서, 센서로부터 센서 데이터를 수신할 때마다 센서 메타데이터를 요청한다는 점이다. 이러한 반복적인 센서 메타데이터 요청은 센서 레지스트리 시스템의 성능을 지속적으로 저하시키는 요인으로 작용한다.

[8]에서는 몇 가지 요인을 고려하여 센서 레지스트리 시스템을 구성하는 방안을 제시하고 있다. 그러나 이 논문은 시스템 구성을 위한 요인 및 구성 방법에 대한 개념만을 제시하고 있다. 무엇보다 기존 센서 레지스트리 시스템과의 정량적

성능 평가 결과를 제공하지 않는다. 따라서 이 논문에서는 요인에 대한 타당성 및 구체적인 처리 연산 구조를 정의하고 하고, 정량 평가를 통해 성능 개선 결과를 보인다.

### 3. 성능 개선 모델

이 절에서는 앞서 기술한 기존 센서 레지스트리 시스템의 문제점을 해결하고 전체적인 성능을 개선할 수 있는 방법에 대하여 기술한다. 이에 앞서, 이 논문에서는 용어의 혼동을 방지하고 논문에 대한 독자의 이해를 돕기 위해 용어를 먼저 정의한다. 이 논문에서 센서로부터 수신한 데이터와 센서 데이터의 의미 정보를 각각 센서 데이터(Sensor Data)와 센서 메타데이터(Sensor Metadata)라고 정의한다. 센서 데이터는 모바일 디바이스가 센서로부터 수신한 실제 값(Sensing Data)을 의미한다. 또한 센서 값과 함께 어떤 센서로부터 수신된 값인지를 나타내는 센서 ID를 포함한다. 따라서 센서 데이터는 센서 값과 센서 ID의 쌍으로 정의된다. 센서 메타데이터는 센서의 유형, 센서로부터 획득한 센서 값의 단위를 포함한 다양한 정보를 포함한다. 마지막으로, 이 논문에서는 센서 데이터와 센서 메타데이터의 집합을 센서 정보(Sensor Information)로서 정의한다.

#### 3.1 성능 향상 모델 개념

앞서 기술하였듯이, [8]에서는 최적화 된 센서 레지스트리 시스템 구성을 위한 몇 가지 요인을 정의하고 있다. 현재 모바일 디바이스는 매우 높은 성능의 다양성을 보이고 있으며, 지속적으로 성능이 향상되고 있다. 따라서 이 논문에서는 [8]에서 정의한 요인 중, 모바일 디바이스 성능과 같은 하드웨어적인 요인은 고려하지 않는다. 또한 이 논문에서는 외부 센서 정보 서비스와의 연동 요인을 고려하지 않는다. 센서 웹 서비스와 같은 외부 정보 서비스의 다양성은 모바일 디바이스의 성능 다양성보다 더욱 복잡하다. 따라서 대표적으로 인정받는 일부 센서 정보 서비스에 대한 정의가 없는 상태에서 이를 요인으로서 포함하는 것은 타당하지 않을 뿐 아니라 현실적으로 불가능

하다.

앞서 기술한 바와 같이, 모바일 디바이스 처리 성능 및 외부 서비스 등의 성능에 중립적이라고 가정할 경우, 센서 레지스트리 시스템의 성능에 영향을 주는 가장 중요한 요인은 센서 데이터에 대한 의미 처리 연산이다. 이 연산은 센서 레지스트리로부터 모바일 디바이스가 센서 메타데이터를 요청하고 원하는 정보를 수신하여 이용하는 과정을 포함한다. 이러한 연산 과정은 센서 레지스트리 시스템에서 모바일 디바이스로 전송되는 센서 메타데이터의 범위와 전송 빈도에 영향을 받는다. 따라서 이 논문에서는 이러한 요인을 고려하여 최적의 연산 처리를 제공하는 기존 센서 레지스트리 시스템의 성능 향상 방법을 정의한다. 즉, 센서 메타데이터의 전송 빈도를 줄이기 위해 센서 메타데이터를 재사용할 수 있도록 한다. 또한 모바일 디바이스에 전송되는 센서 메타데이터의 영역화를 통해 전송되는 정보의 범위를 최적화한다.

### 3.2 성능 향상 모델 정의

앞에서 언급한 바와 같이, 이 논문에서 제안하는 기존 센서 레지스트리 시스템의 성능 향상을 위한 접근 방법의 핵심은 센서 메타데이터의 영역화 및 모바일 디바이스에 전송된 센서 메타데이터의 재사용이다. 센서 메타데이터는 다양한 형태로 영역화가 가능하다. 그러나 이 논문에서는 크게 기본 영역과 응용 영역으로 분류한다.

센서 메타데이터의 기본 영역은 센서로부터 전송된 값의 의미를 해석하기 위한 최소한의 메타데이터 집합을 의미한다. 센서로부터 전송된 센서 데이터의 의미를 인지, 해석, 처리하기 위해서는 전송된 값이 무슨 의미하는지를 이해할 수 있어야 한다. 예를 들어, 임의의 센서 s1으로부터 수신된 센서 데이터를 (s0001, 16)이라고 가정하자. 모바일 디바이스 내의 응용 프로그램이 이 값을 이용하기 위해서는 센서 값 '16'의 의미를 이해해야 한다. 이에 대한 해석을 위해서는 우선 센서 값을 제공한 센서의 유형을 파악해야 한다. 즉, 온도 센서인지, 습도 센서인지를 식별해야 값의 의미를 이해할 수 있다. 만일 센서 s1이 온도 센

서임을 이해하게 되면 온도의 단위, 즉 섭씨인지 화씨인지를 알아야만 센서 값을 정확하고 유효하게 활용할 있다. 이러한 해석 연산을 위해 모바일 디바이스는 센서 레지스트리 시스템에 센서 (ID='s0001')의 유형과 값의 단위에 대한 정보를 요청하게 된다. 이와 같이 센서가 감지한 실제 값을 이해하기 위한 센서 메타데이터를 기본 센서 메타데이터로 분류한다.

센서 데이터를 이용한 임의의 응용 프로그램의 경우, 센서 제조회사, 센서 설치 기관 등 매우 다양한 정보를 활용하는 기능을 제공할 수 있다. 이 논문에서는 이러한 정보를 응용 센서 메타데이터라고 정의한다.

센서 메타데이터 영역화 및 재사용성 지원 여부에 따라 다양한 모델 정의가 가능하다. 이 논문에서는 영역화 및 재사용성 기능 제공 여부에 따라 모델을 표 1과 같이 분류한다.

표 1에서, PM은 영역화 및 재사용성을 지원하지 않는 모델로서, 기존 센서 레지스트리 시스템을 의미한다. EM-1은 전송되는 센서 메타데이터의 영역화 기능을 지원하는 모델로서, 기본 센서 메타데이터를 디폴트로 제공하고 상황에 따라 응용 센서 메타데이터를 제공하는 모델이다. 그러나 재사용성을 제공하지 않는다. EM-2는 영역화 기능을 지원하지 않는 반면, 재사용성을 지원하는 모델이다. 마지막으로, EM-3은 재사용성과 영역화 기능을 지원하는 모델로서, 기본 센서 메타데이터를 디폴트로 전송하고 상황에 따라 응용 센서 메타데이터를 제공한다. 이 모델에서는 제공된 정보를 모바일 디바이스에서 재사용하는 특성을 지닌다.

<표 1> 성능 개선 모델 정의

영역화 재사용	영역화 미지원	영역화 지원
재사용 미지원	PM	EM-1
재사용 지원	EM-2	EM-3

### 3.3 알고리즘 정의

이 절에서는 각 성능 개선 모델의 특성 및 각 모델의 처리 과정에 대하여 서술한다.

첫 번째 모델(EM-1)은 기존 센서 레지스트리 시스템에 영역화 기능을 추가한 모델이다. 센서 레지스트리 시스템에서 전송한 센서 메타데이터

의 재사용성은 고려하지 않고 전송되는 메타데이터의 영역화만을 고려한다. 즉, 기본 센서 메타데이터를 디폴트로 전송하고 모바일 디바이스의 요청이 있을 경우에만 응용 센서 메타데이터를 전송한다. 따라서 센서로부터 수신되는 정보에 대한 지속적인 사용이 드문 경우에 적용할 수 있다.

다음은 EM-1의 전체적인 연산과정을 보여준다.

START\_EM-1 (INPUT s1: Sensor, srs: SRS)

```
sData: SensorData;
sMeta: SensorMetadata;
loop, app: Boolean;
loop←TRUE; app←FALSE;
sID: String;
START_WHILE(loop)
    sData←s1.receieveSensorData();
    sID←sData.getID();
    app←needApplicationMeta();
    START_IF(app)
        sMeta←srs.getMetadata(sID, ALL);
    ELSE
        sMeta←srs.getMetadata(sID, BASIC);
    END_IF
    interpretSensorData(sData, sMeta);
    loop←utilizeSensorInfo(sData, sMeta);
END_WHILE
END_EM-1
```

EM-2는 전송된 센서 메타데이터의 재사용을 통해 모바일 디바이스와 센서 레지스트리 시스템 간의 통신 빈도를 감소시킨다. 즉 반복적인 요청에 따른 센서 레지스트리 시스템의 부하 및 전송에 따른 부하를 감소시킨다. 그러나 EM-2는 센서 메타데이터에 대한 영역화를 지원하지 않는다. 따라서 모바일 디바이스에서 센서에 대한 다양한 정보를 사용자에게 제공하는 서비스에 적합하다.

다음은 EM-2의 전체적인 알고리즘을 보여준다.

START\_EM-2 (INPUT s1: Sensor, srs: SRS)

```
sData: SensorData;
sMeta: SensorMetadata;
got, loop, app: Boolean;
```

```
loop←TRUE; app←FALSE; got←FALSE;
sID: String;
START_WHILE(loop)
    sData←s1.receieveSensorData();
    sID←sData.getID();
    START_IF(!got)
        app←needApplicationMeta();
        sMeta←srs.getMetadata(sID, ALL);
        got←TRUE;
        interpretSensorData(sData, sMeta);
    END_IF
    loop←utilizeSensorInfo(sData, sMeta);
END_WHILE
END_EM-2
```

EM-3은 센서 메타데이터에 대한 영역화 기능 및 재사용 기능을 지원하는 모델이다. 따라서 EM-3은 EM-2와 동일하게 전송된 센서 메타데이터의 재사용을 통해 모바일 디바이스와 센서 레지스트리 시스템 간의 통신 빈도를 감소시킨다. 이는 반복적인 요청에 따른 센서 레지스트리 시스템의 부하 및 전송에 따른 부하를 감소시킨다. 또한 이 모델은 기본 센서 메타데이터를 디폴트로 전송하고 상황에 따라 응용 센서 메타데이터를 제공한다. 따라서 EM-3은 센서 데이터 해석을 위해 필수적인 기본 센서 메타데이터를 제공하고 모바일 디바이스의 요청이나 제공할 필요가 있는 상황에만 응용 센서 메타데이터를 제공한다.

다음은 EM-3의 전체적인 알고리즘을 보여준다.

START\_EM-3 (INPUT s1: Sensor, srs: SRS)

```
sData: SensorData;
sMeta: SensorMetadata;
loop, app, got: Boolean;
loop←TRUE; app←FALSE; got←FALSE;
sID: String;
START_WHILE(loop)
    sData←s1.receieveSensorData();
    sID←sData.getID();
    START_IF(!got)
        app←needApplicationMeta();
    START_IF(app)
```

```

sMeta←srs.getMetadata(sID, ALL);
ELSE
sMeta←srs.getMetadata(sID, BASIC);
got←TRUE;
END_IF
interpretSensorData(sData, sMeta);
END_IF
loop←utilizeSensorInfo(sData, sMeta);
END_WHILE
END_EM-3
    
```

### 4. 성능 평가

#### 4.1 성능 평가 환경 및 평가 모델

기존 방법과 제안 방법과의 성능 평가를 위한 주요 매개변수에 대한 표기법 및 설정 값은 다음과 같다.

- S : 이용하는 센서들의 집합
- n(S) : 이용하는 센서의 개수를 의미하며, 성능 평가를 위한 설정 값은 {10, 20, 30, 40, 50}
- n(SR) : 특정 센서로부터 센서 데이터 수신 횟수로서 설정 값은 {5, 10, 15, 20, 25}
- n(BM) : 기본 센서 메타데이터의 개수로서 평가를 위해 n(BM)=5로 설정
- n(AM) : 응용 센서 메타데이터의 개수로서 성능 평가를 위해 n(AM)=30으로 설정
- t : 단일 센서 메타데이터 획득 시간 (0<m<1)
- AMR : 센서 메타데이터 레지스트리에 대한 응용 센서 메타데이터 요청 비율(Application Metadata Request Ratio)
- BAT : BAT (Basic Metadata Acquisition Time)는 모바일 디바이스에서 SRS로부터 기본 센서 메타데이터를 획득하는 시간
- AAT : AAT(Application Metadata Acquisition Time)는 모바일 디바이스에서 SRS로부터 응용 센서 메타데이터를 획득하는 시간

센서 레지스트리 시스템을 포함한 전체적인 프레임워크의 성능은 센서로부터의 센서 데이터 수신 시간, 센서 메타데이터 전송 시간 및 센서 정

보 이용 시간으로 구분된다. 그러나 센서 레지스트리 시스템의 성능은 센서 메타데이터 전송 시간에만 영향을 받는다. 또한 센서 메타데이터는 기본 센서 메타데이터 전송 시간(BAT)과 응용 센서 메타데이터 전송 시간(AAT)으로 구분된다. 따라서 모든 모델의 성능은 BAT와 AAT의 합으로 정의된다.

기존 방법을 PM이라고 정의할 때, 앞서 기술한 알고리즘을 바탕으로 기존 센서 레지스트리 시스템의 성능 평가를 위해 정의한 PM의 계산 모델 CM(PM)은 다음과 같다.

$$\begin{aligned}
 \text{BAT(PM)} &= n(S) * n(SR) * n(BM) * t \\
 \text{AAT(PM)} &= n(S) * n(SR) * n(AM) * t \\
 \text{CM(PM)} &= \{n(S) * n(SR) * n(BM) * t\} \\
 &\quad + \{n(S) * n(SR) * n(AM) * t\} \\
 &= n(S) * n(SR) * \{n(BM) + n(AM)\} * t
 \end{aligned}$$

이 논문에서 정의한 모델, EM-1, EM-2, EM-3의 계산 모델을 각각 CM(EM-1), CM(EM-2), CM(EM-3)이라고 정의할 때, 각각의 계산 모델은 다음과 같다.

<EM-1의 계산 모델>

$$\begin{aligned}
 \text{BAT(EM-1)} &= n(S) * n(SR) * n(BM) * t \\
 \text{AAT(EM-1)} &= \{n(S) * n(SR) * \text{AMR}\} * n(AM) * t \\
 \text{CM(EM-1)} &= \{n(S) * n(SR) * n(BM) * t\} \\
 &\quad + [\{n(S) * n(SR) * \text{AMR}\} * n(AM) * t] \\
 &= n(S) * n(SR) * \{n(BM) + \text{AMR} * n(AM)\} * t
 \end{aligned}$$

<EM-2의 계산 모델>

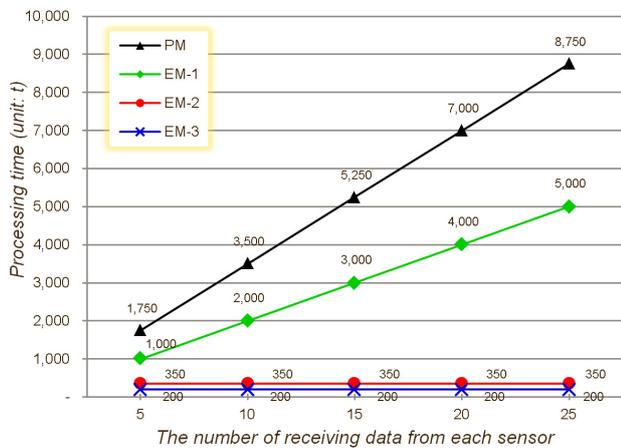
$$\begin{aligned}
 \text{BAT(EM-2)} &= n(S) * n(BM) * t \\
 \text{AAT(EM-2)} &= n(S) * n(AM) * t \\
 \text{CM(EM-2)} &= \{n(S) * n(BM) * t\} + \{n(S) * n(AM) * t\} \\
 &= n(S) * \{n(BM) + n(AM)\} * t
 \end{aligned}$$

<EM-3의 계산 모델>

$$\begin{aligned}
 \text{BAT(EM-3)} &= n(S) * n(BM) * t \\
 \text{AAT(EM-3)} &= n(S) * \{n(S) * \text{AMR}\} * n(AM) * t \\
 \text{CM(EM-3)} &= n(S) * \{n(BM) * t\} \\
 &\quad + \{n(S) * \text{AMR} * n(AM) * t\} \\
 &= n(S) * [n(BM) + \{\text{AMR} * n(AM)\}] * t
 \end{aligned}$$

### 4.2 성능 평가 결과

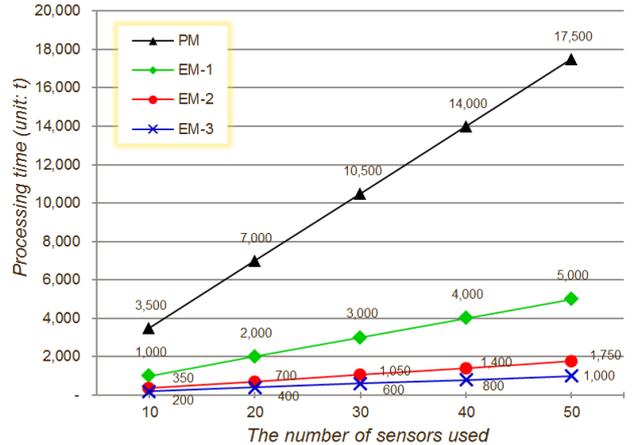
그림 1은 특정 센서로부터의 센서 데이터 수신 횟수인  $n(SR)$ 의 변화에 따른 성능 평가 결과를 보여준다. 이 성능 평가에서, 센서의 개수인  $n(S)$ 와 응용 센서 메타데이터 요청 비율(Application Metadata Request Ratio, AMR)을 각각 '10'과 '50%'로 설정하였다. 그림 1에서 알 수 있듯이, 기존 모델인 PM과 센서 메타데이터 영역화를 통해 PM 성능을 개선한 EM-1은  $n(SR)$  값과  $n(S)$ 의 값에 영향을 받는다. 따라서  $n(SR)$ 의 값이 증가할수록 처리 시간 또한 증가하게 된다. 반면 EM-2와 EM-3은  $n(SR)$  값에 영향을 받지 않으며, 따라서 센서의 개수가 고정적일 경우,  $n(SR)$  값에 무관하게 EM-2와 EM-3은 각각 일정한 성능을 보였다. 평가 결과에서, PM에 비해 EM-1은 약 7배의 성능 개선이 이루어졌으며, 가장 높은 성능을 보인 EM-3은 PM에 비해 매우 높은 성능 향상을 보였다.



<그림 1> 센서 수신 횟수에 따른 성능 평가 결과

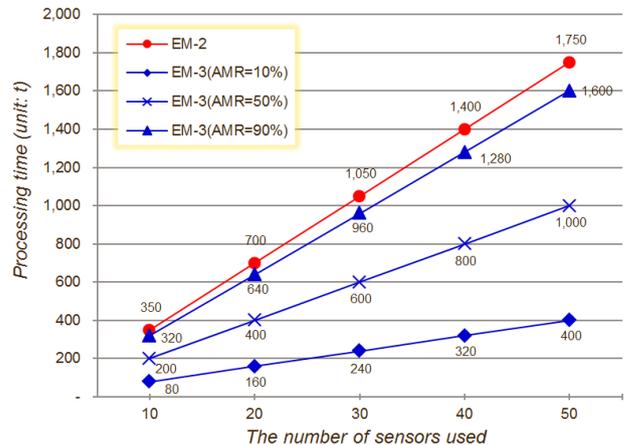
그림 2는 임의의 센서로부터 센서 데이터를 수신하는 횟수인  $n(SR)$ 을 상수로 정의하고 센서 개수인  $n(S)$  값을 변수로 정의한 평가 결과를 보여준다. 즉,  $n(S)=10$ ,  $AMR=50\%$ 로 각각 설정하고 센서 개수에 따른 성능 평가를 수행하였다.  $n(SR)$ 을 상수로 정의한 평가 결과와 달리, PM과 EM-1을 포함한 EM-2와 EM-3, 두 모델은 센서의 개수가 증가할수록 처리 비용이 증가함을 알 수 있다. 또한 EM-2고 EM-3과의 비교 평가에서,

센서의 개수 증가에 따라 PM이나 EM-1의 처리 비용 격차가 지속적으로 증가하고 있음을 알 수 있다. 특히 기존 모델과 EM-3은 현격한 성능 차이를 보임을 알 수 있으며, PM과 비교하여 EM-3은 매우 높은 성능 개선 결과를 보였다.



<그림 2> 센서 개수에 따른 성능 평가 결과

그림 3은 AMR의 비율에 따른 EM-2와 EM-3의 성능 차이를 확인하기 위한 평가 결과이다.



<그림 3> 응용 센서 메타데이터 요청 비율에 따른 평가 결과

성능 평가 결과에서, EM-2는 디폴트로 기본 센서 메타데이터와 응용 센서 메타데이터를 요청하기 때문에 AMR 비율과는 무관하게 일정한 결과를 보인다. 그러나 EM-3은 디폴트로 기본 센서 메타데이터를 요청하고 필요시 응용 센서 메타데이터를 요청한다. 따라서 AMR의 비율에 따라 각

각 다른 결과를 보인다. 그림 3에서, AMR 비율이 '10%'인 경우, EM-2와 EM-3의 처리 성능은 약 4.4:1.0으로서 EM-3이 4배 이상의 우수한 성능을 보였다. 또한 AMR 비율이 '90%'인 경우에도 EM-3이 약 1.1배의 나은 성능을 보였다..

### 5. 결론 및 향후 연구

이 논문에서는 기존 센서 레지스트리 시스템의 성능 개선 방법에 대하여 기술하였다. 성능 개선을 위한 핵심 요소는 데이터의 영역화 및 재사용성으로서, 각각의 지원 여부에 따른 모델을 분류하였다. 또한 각 성능 향상 모델의 특징과 처리 알고리즘을 보였으며, 성능 분석을 통해 기존 방법과 제안 방법과의 비교 평가를 수행하였다. 성능 평가 결과에서, 이 논문에서 제안한 방법이 기존 방법에 비해 우수함을 확인하였으며, 이용하는 센서의 개수와 특정 센서로부터의 데이터 수신 횟수가 증가할수록 처리 성능 차이가 심화됨을 알 수 있었다.

향후 연구로서, 보다 지능적인 적응형 센서 데이터 활용을 위한 센서 레지스트리 시스템 개발이 요구된다. 즉, 문맥(상황) 인지 기반의 센서 레지스트리 시스템 개발이 요구된다. 이를 위해서는 센서 레지스트리 시스템의 기능 확장은 물론 의미 처리의 주체인 모바일 디바이스에서의 기능 확장에 대한 연구가 요구된다. 아울러 사용자의 이용 패턴 및 이동 패턴 기반의 예측을 통한 성능 개선에 대한 연구가 요구된다.

### 참 고 문 헌

[1] Sheth, A., Henson, C., & Sahoo, S. S. (2008). Semantic Sensor Web. *IEEE Internet Computing*, 12(4), 78-83.

[2] Goodwin C. & Russomanno, D. J. (2006). An Ontology-based Sensor Network Prototype Environment. *Proc. of the 5th International Conference on Information Processing in Sensor Networks*, 1-2.

[3] Bouillet, E., Feblowitz, M., Liu, Z., Ranganathan, A., Riabov, A., Ye, F. (2007).

A Semantics-based Middleware for Utilizing. *Heterogeneous Sensor Networks. Lecture Notes in Computer Science*, 4549, 174-188.

[4] Xue, G., Pan, Q., Li, M. L. (2007), A New Semantic-based Query Processing Architecture. *Proc. of the 2007 International Conference on Parallel Processing Workshops*, 63.

[5] H. Jeung, H., Sarni, S., Paparrizos, I., Sathe, S., Aberer, K., Dawes, N., Papaioannou, T. G., & Lehning, M. (2010). Effective Metadata Management in Federated Sensor Networks, *Proc. of the 2010 IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing*, 107-114.

[6] Ibrahim, I. K., Kronsteiner, R., & Kotsis, G. (2005). A Semantic Solution for Data Integration in Mixed Sensor Networks, *Computer Communications*, 28(13), 1564-1574.

[7] Jeong D. & Ji J. (2011). A Registration and Management System for Consistently Interpreting Semantics of Sensor Information in Heterogeneous Sensor Network Environments. *Journal of KIISE: Databases*, 38(5), 289-302.

[8] Jeong, D., Doo, M., & Park, H. (2012). A Study of the Alternative Sensor Registry System Architectures. *Proc. of the 2012 KSCI Winter Conference*, 20(1), 109-112.



### 정 동 원

1997 군산대학교  
전자계산학과(이학사)

1999 충북대학교  
전자계산학과(이학석사)

2004 고려대학교 컴퓨터학과(이학박사)

2005~현재 군산대학교 통계컴퓨터학과(교수)

관심분야: 데이터베이스, 의미 교환 및 공유 분야

E-Mail: djeong@kunsan.ac.kr