

IoT 데이터 기반 센싱 데이터 수집 환경

¹홍명우, *²강윤희

¹우송정보대학, mwhong@wsi.ac.kr

².교신저자 백석대학교, yhkang@bu.ac.kr

IoT based Data centric sensory data aggregation environment

¹Hong Myoung Woo, ²Yunhee Kang

¹Woosong College, mwhong@wsi.ac.kr

*², Corresponding Author Baekseok University, yhkang@bu.ac.kr

요약

IoT 환경에서 수집 데이터 관리의 주요기능은 실시간 데이터 획득을 기반으로 잘 정의된 명세에 따라 센서로부터 외부환경의 센싱 자료를 수집한다. 수집된 센싱 자료는 비즈니스 의사결정 활용에 필수적이다. 데이터 수집의 자동화는 다양한 센싱 정보에 대한 체계적 표현과 수집 메커니즘이 요구된다. 본 논문에서는 통합 센서 표현을 위한 CDM(Common Device Model) 공통장치 모델 기반의 SML을 제안한다. SML은 IoT 장치의 상세명세를 포함한 정적정보 및 센싱 값에 대한 수행과정의 동적정보를 기술하기 위한 메타데이터를 제공한다. MQTT 프로토콜 기반의 데이터 수집 환경을 기술한다.

Abstract

The main function of data aggregation management in IoT environment is to collect sensory data based on real-time data acquisition from sensors well-defined in outside environment. It is necessary that the collected sensory data is used for making a business decision. Automation of data aggregation is required to represent sensors systematically and data aggregation mechanism. In this paper, we propose SML based on CDM(Common Device Model) that is a data model for an integrated sensor with diverse sensors. SML represents metadata related with details of specification about static information of a sensor and sensor data from sensor data in runtime. We also present MQTT protocol based data aggregation environment.

Keywords: IoT, Data aggregation, CDM, SML, MQTT

I. 서론

IoT 용어는 1999 년 MIT Auto-ID 에서 제안되었으며, RFID 관련 공동연구 프로젝트인 Casagras 는 데이터 캡처 및 통신 기능의 가용성을 활용해 물리적 객체 및 가상 객체를 연결하는 글로벌 네트워크 인프라로 정의하였다[7]. 최근 다양한 센서 및 측정장치의 개발 및 저렴한 가격으로 IoT 를 포함한 많은 분야에서 이들의 보급이 이루어짐에 따라 센싱 데이터 활용이 지속적으로 증가하고 있다[1,2].

* Corresponding Author

Received: Mar. 19, 2019, Revised: Apr. 02, 2019, Accepted: Apr. 05, 2019

IoT 환경의 자료처리는 빅데이터의 세가지 특성인 데이터 다양성(variety), 데이터 크기(volume) 및 데이터 증가 속도(velocity)를 갖는다. IoT 환경의 생태계는 각종 센서 및 측정장치에서 수집된 데이터로 인해 빅데이터 처리를 요구한다.

IoT 환경에서 수집데이터 관리의 주요기능은 실시간 데이터 획득을 기반으로 잘 정의된 명세에 따라 센서로부터 외부환경의 센싱 자료를 수집한다. IoT 는 인터넷 연결이 가능하고 각종 센서를 탑재한 디바이스간 커뮤니케이션이 가능하여야 하며, 생성되어진 정보를 기반으로 새로운 서비스 또는 가치 창출이 가능하여야 하며, 수집된 센싱 자료는 비즈니스 의사결정 활용에 필수적이다. 데이터 수집의 자동화는 다양한 센싱 정보에 대한 체계적 표현과 수집 메커니즘이 요구된다[3,4].

본 논문에서는 통합 센서 표현을 위한 CDM(Common Device Model, 공통장치 모델) 기반의 SML 을 제안한다. SML 은 IoT 장치의 상세명세를 포함한 정적 정보 및 센싱 값에 대한 수행과정의 동적정보를 기술하기 위한 메타데이터를 제공한다. MQTT 프로토콜 기반의 데이터 수집 환경을 기술한다.

II. 관련연구

IoT 환경은 무선 센서 네트워크(Wireless Sensor Network)과 RFID 의 효율적인 통합에서 시작하였으며, 이러한 통합 환경에서 통신하는 사물의 범위를 넓혀 인간과 사물, 서비스 세 가지 분산된 환경 요소들이 인터넷 범위로 포함되어 센싱, 네트워킹, 정보 처리 등의 상호작용을 한다. 사용자 측면에서 스마트 기기는 자체 내장된 센서를 활용하여 새로운 응용 서비스를 제공하고 있으며, 이를 위한 프레임워크에 대한 요구가 증가하고 있다.

스트리밍 기반 센서데이터 운영 환경에서 수집된 센서 데이터를 응용에 전달하기 위해서는 파이프-필터(Pipe-and-Filter) 패턴을 이용한다. 파이프-필터 패턴에서 각 컴포넌트는 입력과 출력 집합을 갖는다. 컴포넌트는 입력단(input end-point)에서 데이터 스트림을 읽고 출력단(output end-point)에 데이터 스트림을 출력한다. 필터는 데이터 스트림 변환에 사용하며, 변환 결과는 이를 필요로 하는 컴포넌트에 전달한다[8].

2013 년 OASIS(Organization for the Advancement of Structured Information Standards)에 의해 표준화된 MQTT 는 사물인터넷과 같이 자원이 제한된 네트워크 환경에 최적화된 경량 메시지 전송 프로토콜이다. MQTT 는 다수의 사용자, 디바이스를 지원하며 응용 코드 변경 없이 기능을 사용할 수 있기 때문에 다른 사물인터넷 응용 프로토콜 보다 유연성과 확장성, 그리고 신뢰성이 뛰어나다[5]. MQTT 는 IoT 환경에서 활용이 용이한 다음의 특징을 갖는다.

- 경량의 Publisher/Subscriber(Pub/Sub) 메시징 프로토콜
- M2M 와 IoT 에서의 사용
- 저전력, 신뢰할 수 없는 네트워크, No TCP/IP 기반에서 운용

Figure 1.은 MQTT 의 메시징을 위한 모델인 Publisher/Subscriber 모델을 보인 것이다. MQTT 는 게시자와 구독자 사이에서 중계 역할을 하는 Broker 가 필요하다. Broker 는 MQTT Message 를 중계하는 물리적인 서버이다. MQTT 의 토픽은 /(slash)로 구분된 계층적 구조를 갖는다. MQTT 는 발간을 할 때는 보내고자 하는 토픽과 데이터를 전송하고 구독을 할 때는 해당 토픽으로 오는 정보가 있는지 여부를 확인하고 구독한다.

MQTT 프로토콜은 요청/응답 패턴이나 이벤트 구독패턴 대신 게시(publish)/구독(subscribe) 패턴을 기반으로 동작한다. 게시/구독 패턴은 세가지 개체로 구성된다.

- 게시자 publisher : 메시지 중개자에 연결하여 콘텐츠를 게시한다
- 구독자 subscriber : 게시자처럼 메시지 브로커에 접속하여 원하는 내용을 구독한다
- 메시지 중개자 message broker : 게시된 콘텐츠를 구독자에게 전달한다.

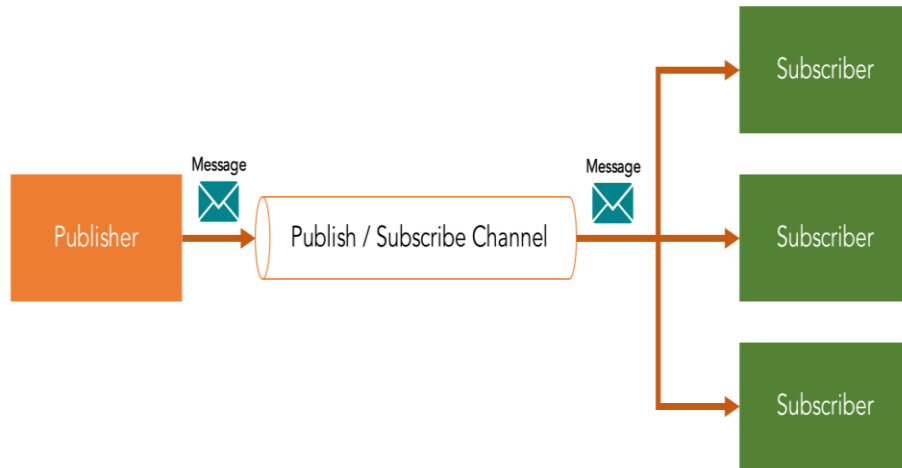


Figure 1. Publisher/Subscribe Model

Ⅲ. 데이터 수집 시스템

3.1 아키텍처 설계

IoT 서비스 플랫폼의 소프트웨어 아키텍처는 다음의 서비스 시나리오를 지원하여야 한다.

- IoT는 자체적인 사물에 센싱과 같은 기능을 내장시키고 이들 간에 통신을 함으로써 사용자의 행태 및 상황을 인식하여 신규의 가치를 창출하는 것을 목적으로 한다.
- IoT 환경에서 응용은 네트워크에 연결된 여러 디바이스들을 이용하여 사용자에게 유용한 정보와 편의를 제공할 수 있다.
- IoT 디바이스는 단순한 데이터를 제공하기도 하며, 변환을 통해 응용에 다양한 서비스를 제공한다.
- IoT 응용은 유무선 통신, 센서 등의 다양한 IoT 기술을 적용하며 서비스 지향 환경 구축한다.

서비스 시나리오 운영을 위해서는 사물에서 일어나는 모든 데이터를 적기에 수집하고 분석 및 예측하여 보다 정교하고 유용한 정보를 사용자에게 제공하여야 하며 이를 위한 플랫폼의 주요 요구기능은 다음과 같다.

- IoT 장치 등록을 위한 프로파일 관리
- IoT 사용자 프로파일 관리
- IoT 서비스 및 응용 프로파일 관리

Figure 2.의 설계된 아키텍처는 테스트베드 관리 서버, IoT 장비 핸들러 및 IoT 장치 레지스트리로 구성된다. 테스트베드 관리 서버는 MQTT 와 같은 Pub/Sub 메커니즘에서 구독자로서 역할을 수행하고 센싱 데이터와 예외, 오류 및 결함 등의 센서 상태를 전달받는다. IoT 장치 핸들러는 게시자의 역할을 수행하며 이벤트 계산시스템으로 동작한다. 이는 입력 받은 데이터 소스인 센서로부터 수집된 센싱 데이터의 필터링(filtering)을 수행한다[6]. IoT 장치 핸들러에서 다루는 소프트웨어 센서는 SML에 따라 기술하며 테스트베드 서버는 전달된 센싱 데이터의 유효성 검증을 위해 해당 명세를 요청한다.

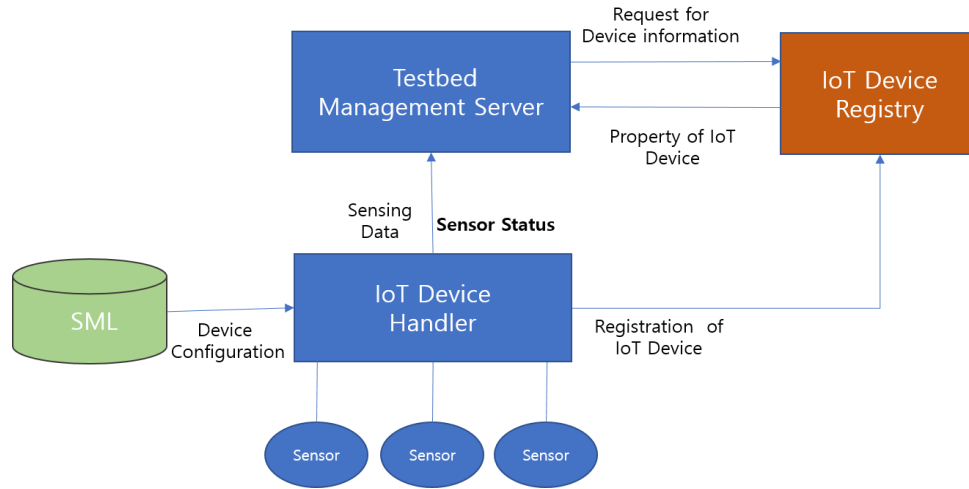


Figure 2. Software architecture for notification-based data aggregation testbed

3.2 CDM

IoT 네트워크는 사물(객체)들이 주어진 시간 및 공간에서 연결되어 구성되며 분석을 통해 사용자에게 의해 활용된다. 이들 사물들은 네트워크에 접속하여 정보를 교환하고 정보를 처리한다. 사물을 표현하는 객체는 네트워크 기능을 갖는 통합 센서의 형태로서 변환되며, 통합센서는 센서, 액추에이터 및 처리 장치로서 구성된다. CDM은 통합센서를 갖는 IoT 장치의 상세명세를 포함한 정적 정보 및 센싱 값에 대한 동적정보를 기술하기 위한 메타데이터를 제공한다. 센싱 데이터의 수집 및 전송량은 CDM 메타데이터의 동적정보의 데이터 수집 계획(Data collection plan)을 기반으로 정의된다. 전체 네트워크 처리율을 반영하여 IoT 장치의 데이터 전송율을 동적으로 조정한다.

이 실험에서는 가상 센서의 관리를 위해 CDM를 기반으로 IoT 장치를 등록한다. 이를 위해 IoT 장치의 식별자를 읽어 이를 IoT 장치 식별자로 등록한다. 각 IoT 장치는 신뢰성을 검증하기 전까지 임의의 사용시간 TTL을 부여 받아 센서의 수집정보를 통합 게이트웨이를 통해 수집 및 분석을 위한 클라우드 플랫폼에 전송하여 유지하도록 한다.

3.3 실험 환경 및 결과

구성된 실험환경은 데이터 수집을 위한 CDM을 기반으로 SML(Sensor Markup Language)를 설계하였다. Figure 3.은 전체적인 소프트웨어 센서의 전체 구조를 보인 것으로 소프트웨어 센서를 1개 이상의 물리 센서로 구성되며 SML로 기술한다.

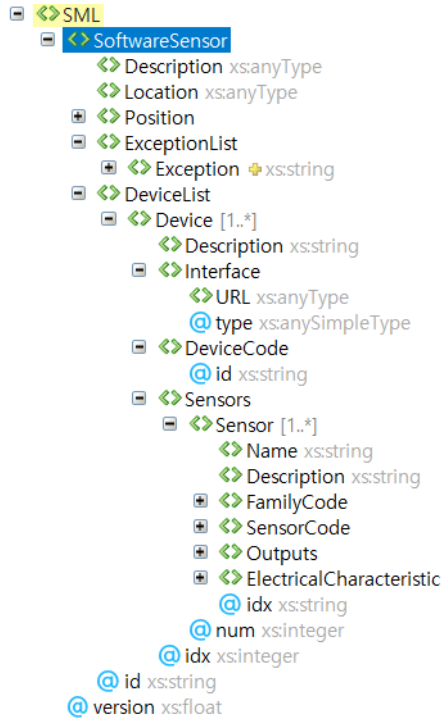


Figure 3. Overall integrated sensor structure

정의된 SML 에서 소프트웨어 센서의 이름과 설명을 엘리먼트 Name 과 엘리먼트 Description 을 통해 기술하며 구성된 물리센서에 대한 정보는 엘리먼트 Sensor 에 기술한다. 엘리먼트 Sensor 는 측정하는 값의 범위 및 해상도를 기술한다.

센싱 데이터의 수집 체계 정의는 node-red 를 활용하며, 센싱 데이터를 JSON 메시지로 구성하여 전송한다. Node-RED 는 IBM 의 Emerging Technology Services 팀이 개발한 흐름 기반 프로그래밍 도구(flow-based programming tool)입니다. 기본적으로 자바스크립트(JavaScript) 언어를 제공한다. Pub/sub 데이터 전송을 위해 MQTT 미들웨어인 Mosquitto 버전 1.4.10 를 설치하여 사용한다. 라즈베리 파이에는 센싱 데이터 출판을 위한 Python 응용을 구성하며 이를 위해 mqtt-paho 패키지를 사용한다. 출판된 센싱 정보는 클라우드의 리눅스 서버에서 수집한다. 구독자 응용 역시 Paho 패키지를 사용하여 구성한다. Figure 4.는 node-red 를 통해 구독된 온도 및 습도 정보의 JSON 형식 데이터를 보인 것이다.

```

2/2/2017, 1:01:34 PM node: Output
Home/Univer/Temp : msg.payload : Object
> { hostname: "raspberrypi", name: "Temperature", value: 20.399999618530273 }

2/2/2017, 1:01:37 PM node: Output
Home/Univer/Humi : msg.payload : Object
> { hostname: "raspberrypi", name: "Humidity", value: 45.29999923706055 }

2/2/2017, 1:01:37 PM node: Output
Home/Univer/Temp : msg.payload : Object
> { hostname: "raspberrypi", name: "Temperature", value: 20.399999618530273 }
  
```

Figure 4. subscribed sensory data with temperature and humidity in JSON

IV. 결론

이 논문에서는 실시간 데이터를 기반으로 한 외부환경의 센싱 자료에 대한 데이터 기반 관리를 위한 소프트웨어 아키텍처를 설계하여 데이터수집 환경을 구성하였다. 설계된 환경에서는 MQTT 프로토콜을 기반으로 온도 및 습도 데이터를 획득하였다. 온도 측정을 위한 소프트웨어 센서는 CDML 을 참고하여 SML 로 기술하였으며, 이를 기반으로 수집된 센싱 데이터의 유효성 및 상태에 따른 이벤트 처리를 수행한다. 자동화된 테스트 환경에서 사용자의 높은 운영 편의성을 위해 node-red 을 사용하여 수집하였으며, 결과 자료는 비즈니스 의사결정에 활용할 수 있다.

V. 참고문헌

- [1] Zuehlke, Detlef. "Smartfactory—from vision to reality in factory technologies," in IFAC Proceedings, 2008, pp.14101-14108
- [2] May, G.S., Spanos, C.J.: Fundamentals of Semiconductor Manufacturing and Process Control. In *John Wiley & Sons*, 2006
- [3] L. D. Xu, W. He, and S. Li, "Internet of Things in Industries: A Survey," *IEEE Trans. Industrial Informatics*, Vol. 10, No. 4, pp. 2233–2243, 2014.
- [4] Chenxi, H. A Study of Publish/Subscribe Systems for Real-Time Grid Monitoring. IEEE International Parallel and Distributed Processing Symposium, 2007
- [5] Patrick Th, E., Rachid, G., Joe, S.: Distributed Asynchronous Collections: Abstractions for Publish/Subscribe Interaction. Proceedings of the 14th European Conference on Object-Oriented Programming, pp. 252-276, Springer-Verlag , 2000
- [6] François, A.R.J.: Software Architecture for Computer Vision: Beyond Pipes and Filters. Prentice Hall, 2003
- [7] Ashton, K. 1999, June 22. That 'Internet of Things' thing. RFID Journal.
- [8] Kaiwen Zhang, Tilmann Rabl, Yi Ping Sun, Rushab Kumar, Nayeem Zen, and Hans-Arno Jacobsen, PSBench: A Benchmark for Content- and Topic-based Publish/Subscribe Systems, Middleware Demos, 2014