

# Asynchronous Sensing Data Aggregation and Processing Mechanism for Internet of Things Environment

Yunhee Kang<sup>†</sup> · Wan-Ki Ko<sup>††</sup>

## ABSTRACT

In the Internet of Things(IoT) era, many of the things or objects that enclose our environments are able to associate with those things on the Internet. To construct IoT systems, it needs to consider a component for acquiring and aggregating of sensory data via things with sensors and instruments, which is connected by diverse networks, in IoT environment. An IoT system is intrinsically distributed in a variety of ways. In addition, to manage an IoT system efficiently, interoperability is needed to meet requirements while the IoT system is designed to deliver data among its applications. In this paper, a reference architecture based on asynchronous messaging is defined and used for designing an IoT system. To apply the architecture, we discuss how to manage data streams with real-time characteristics and make a prototype based on pipe-and-filter to produce and consume them by a pub/sub messaging system NaradaBrokering.

**Keywords :** IoT, Message Based Software Architecture, Pub/Sub Messaging System, Data Aggregation

## 사물 인터넷 환경에서 비동기 센싱 데이터 수집 및 처리 메커니즘

강 윤 희<sup>†</sup> · 고 완 기<sup>††</sup>

### 요 약

Internet of Things(IoT) 시대에 우리 생활 주변의 많은 사물 또는 객체가 인터넷상에서 연관될 수 있다. IoT 시스템의 구축을 위해서는 다양한 네트워크로 연결된 센서 및 계측장치를 갖는 사물을 통해 IoT 환경의 센싱 데이터를 획득하고 수집을 위한 컴포넌트 개발이 필요하다. 이러한 IoT 시스템은 다양한 형태로 분산되며, 추가적으로 이러한 시스템의 효율적인 관리를 위해서는 애플리케이션 간의 자료전달을 위한 요구사항으로 상호운영성을 만족해야 한다. 이 논문에서는 비동기 기반 메시징 시스템을 기반으로 한 IoT 시스템의 참조 아키텍처를 설계한다. 참조 아키텍처는 IoT 시스템을 설계과정에서 사용한다. 여기서는 참조 아키텍처의 적용을 위한 실시간 특성의 데이터 스트림 관리 방법을 논의한다. 그리고 이들 데이터의 생산 및 소비를 위한 파이프-필터 기반 프로토타입을 pub/sub 메시징 시스템인 NaradaBrokering을 활용하여 구성하는 방법을 논의한다.

**키워드 :** 사물 인터넷, 메시지 기반 소프트웨어 아키텍처, Pub/Sub 메시징 시스템, 데이터 수집

### 1. 서 론

사물이 인터넷에 연결되어 정보를 공유하는 것으로 정의되는 사물 인터넷(IoT, Internet of Things)에서는 다양한 센서의 보급으로 IoT 환경이 빠르게 구축됨에 따라 인터넷에 연결된 사물 또는 객체로부터 센싱 데이터가 수집되고 있다[1, 2]. 센싱 데이터 수집 시스템은 사물과 연관된 센서로부터 수집된 정보를 가공 및 처리하기 위해 서비스 지향(service oriented) 시스템 설계가 요구된다. 이러한 시스템

설계는 IoT 환경에서 사물이 지능화되어 인터넷상의 사용자에게 주변의 다양한 정보를 제공함으로써 새로운 서비스 생태계 형성을 용이하게 한다[3].

IoT 시스템의 구성요소는 물리적으로 분산되며, 이러한 시스템의 효율적인 관리를 위해서는 시스템 설계 과정에서 애플리케이션 간의 자료 교환을 위한 비기능성 요구사항(non-functional requirement)이 고려되어야 한다. 사용자에게 필요한 서비스 제공을 위해서는 IoT 환경에서 유무선의 센서 네트워크(sensor network)와의 연동을 통해 외부에 위치하고 있는 다양한 센서 정보를 수집 또는 제어하여 애플리케이션에 정보를 제공할 수 있는 공통 플랫폼(common platform)이 필요하다. IoT 공통 플랫폼 설계에서는 사물들과 애플리케이션의 원활한 인터페이스를 지원하기 위한 아키텍처가 요구된다.

<sup>†</sup> 정 회 원 : 백석대학교 정보통신학부 부교수

<sup>††</sup> 정 회 원 : 제주한라대학교 방송영상과 부교수

Manuscript Received : September 11, 2014

Accepted : October 13, 2014

\* Corresponding Author : Yunhee Kang(yhkang@bu.ac.kr)

본 논문에서는 센싱 데이터 수집과 수집된 데이터의 애플리케이션 처리를 위한 참조 아키텍처를 설계한다. 설계된 아키텍처는 스트림 기반의 분산 처리를 위한 데이터 활용을 목적으로 한다. 이를 위해 설계된 아키텍처는 pub/sub 기반 메시징 미들웨어를 활용하여 IoT에서 요구되는 사건(Event), 메시지(Message) 및 애플리케이션(Application) 간의 상호운용성(interoperability) 및 확장성(scalability) 문제를 해결하기 위한 서비스로 개발한다.

이 논문의 2절에서는 IoT의 환경 특징 및 웹 기반 통합을 위한 접근인 WoT(Web of Things)와 모바일 환경을 위한 프레임워크인 Reflex 프로젝트와 ODK를 기술한다. 3절에서는 비동기 센싱 데이터 수집 IoT 아키텍처 설계와 설계된 아키텍처의 검증에 관한 실험 환경 및 결과를 기술한다. 마지막으로 4절에서는 결론을 기술한다.

## 2. 관련 연구

IoT는 자체적인 사물에 센싱과 같은 기능을 내장시키고 이들 간에 통신을 함으로써 사용자의 행태 및 상황을 인식하여 신규의 가치를 창출하는 것을 목적으로 한다. IoT 환경에서 애플리케이션은 네트워크에 연결된 여러 디바이스들을 이용하여 사용자에게 유용한 정보와 편의를 제공할 수 있다. IoT 디바이스는 단순한 데이터를 제공하기도 하며, 변환을 통해 애플리케이션에 다양한 서비스를 제공한다. 이러한 서비스 지향 환경 구축은 유무선 통신, 센서 등의 다양한 IoT 기술을 적용한다[4].

IoT 환경은 무선 센서 네트워크(Wireless Sensor Network)와 RFID(Radio Frequency Identification)의 효율적인 통합과 관련되며, 이러한 통합 환경에서 통신하는 사물의 범위를 넓혀 인간과 사물, 서비스의 세 가지 분산된 환경 요소들이 인터넷 범위로 확장되어 센싱, 네트워킹, 정보 처리 등의 상호작용을 통해 관계를 갖는다[2, 3].

웹은 인터넷을 통해 정보자원의 접근이 용이한 환경으로서 IoT의 사물에 대한 접근성을 높이기 위해 모든 사물을 웹으로 통일하고자 하는 Web of Things(WoT)를 제시하였다[5]. WoT에서는 스마트 게이트웨이를 통하여 센서, 장치와 액추에이터들을 객체로 웹의 주소체계를 기반으로 표현하고 이를 등록하도록 함으로써 궁극적으로 물리적 사물을 웹 인터페이스를 통해 검색하고 제어하고 조합한다. 이들 간의 통신은 REST를 기반으로 하며 물리적 매시업(mash-up)까지 지원한다. 개념적으로 WoT는 기존의 객체 식별을 위해 사용하고 있는 RFID의 확장으로서 인터넷 기반 연계를 위한 주소체계를 구성한다.

Reflex 프로젝트는 스마트폰 애플리케이션과 외부 센서와의 인터페이스의 차이로 인해 발생하는 프로그래밍의 어려움을 해결하기 위해 별도의 데이터 센싱을 위한 인터페이스를 사용하여 이형 스마트폰의 투명한 프로그래밍을 지원하는 프레임워크를 제안한다[6].

모바일 환경에서 외부 센서로부터 수집된 정보를 스마트 기기에서 활용하도록 지원하기 위한 ODK(Open Data Kit) 센서 프레임워크는 센싱 데이터의 수집 및 모니터링 애플리케이션을 위해 센서 통합과정을 단순화하기 위해 설계되었으며, 센서 관리(sensor managing) 계층을 두어 외부로부터 유입되는 센서 정보를 저장하고 관리한다[10].

## 3. 비동기 센싱 데이터 수집 IoT 아키텍처 설계

### 3.1 IoT 개념 프레임워크

IoT 환경에서는 사물과 사람의 상호작용이 이루어지며, 이 과정에서 다양한 서비스를 지원받게 된다[5]. 다양한 서비스와 콘텐츠는 상호 연동을 통해 운영되며, 사용자의 디바이스와 사물은 서비스로 구성된 애플리케이션 수행을 위해 메시징 서비스를 통해 정보를 교환한다. Fig. 1은 설계된 IoT 개념 프레임워크이다.

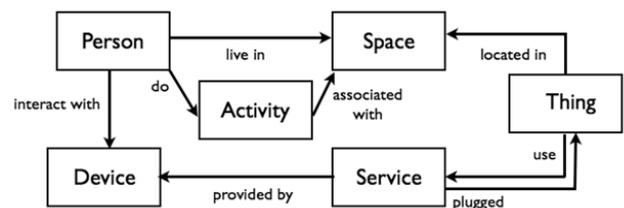


Fig. 1. The designed IoT conceptual framework

설계된 IoT 개념 프레임워크에서 IoT 환경은 다수의 Space로 구성되며, 단위 Space는 IoT의 컨텍스트(context) 구조를 운영한다. Person은 Activity에 참여하며, Space에서 여러 Activity를 수행한다. Thing은 센서와 관계하며, Space에 위치한다. Person은 스마트폰과 같은 Device를 사용하며, 응용에 필요한 Service를 제공받는다. Service는 물리 또는 가상 Thing에도 제공된다. Person의 개인 Device는 Service의 제공을 통해 Thing과의 상호작용을 위한 인터페이스로서 서비스 프로파일(service profile)을 사용한다.

IoT 개념 프레임워크에서 서비스는 사용자에게 특정한 기능을 제공하는 구성요소로서 소프트웨어 기본 구성단위이다. 즉, 서비스는 애플리케이션에서 공통적으로 필요로 하는 소프트웨어의 부분이다. 애플리케이션은 특정한 도메인을 위해서만 활용되는 큰 범위의 소프트웨어라고 할 수 있다. Fig. 2는 애플리케이션과 서비스의 관계를 표현한 것이다.

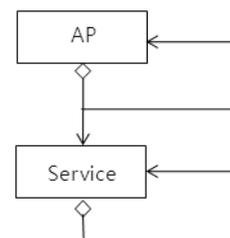


Fig. 2. The relationship between application and service

- 애플리케이션은 서비스를 포함할 수 있다.
- 애플리케이션은 애플리케이션을 포함할 수 있다.
- 서비스는 서비스를 포함할 수 있다.

IoT 개념 프레임워크에서는 수집정보의 분석을 통해 컨텍스트를 분석하고 컨텍스트에서 메시지의 상호운영을 개념화하여 정의한다. 컨텍스트는 1개 이상의 상황으로 표현된다. IoT 환경에서 센서에 의한 통지 또는 사람에 의한 통지 등 시스템 외부에서는 환경 변화에 의해 메시지를 발생한다.

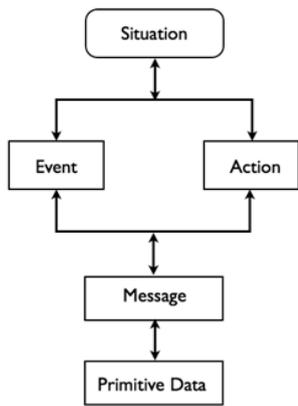


Fig. 3. Context structure of IoT

Fig. 3은 IoT 컨텍스트 구조를 보인 것으로 메시지는 센서의 원시 데이터(primitive data)와 구분되고 센서로부터 수집된 세부사항을 은닉할 수 있도록 한다. 원시 데이터는 센서의 유형에 따라 특별한 자료 형식과 표현값을 표현하며 메시지는 원시 데이터에 대한 1차 해석 후의 추상된 값으로 조정된다. 이벤트(Event)는 모니터링이 필요한 사건으로서 수신된 메시지의 해석으로 표현된다. 행위(Action)는 이벤트에 의해 트리거(trigger) 되는 조치로서 장치에 대한 제어에

사용한다. 메시지는 다수의 메시지의 합성을 통해 새로운 메시지로 구성한다. 상황(Situation)은 복수의 이벤트나 행위에 의해 표현되며 서비스의 상태를 변경한다.

IoT의 컨텍스트의 상세설계인 Context 정보는 지속적으로 변화하는 상태의 정보를 수집하고 분석하고 관리될 수 있다. 상황정보는 하나의 엔티티에 의해서만 결정되지는 않으며, 다양한 개체 또는 활동의 상태에 의해 종합적으로 분석되어야 한다. Fig. 4는 상황정보를 표현하는 Context 클래스의 예를 보인 것이다. 이 예에서 Context는 위치(LocationContext), 활동(ActivityContext), 서비스 또는 디바이스(ServiceContext), 사용자(UserContext)에 의해 표현된다.

3.2 Pub/sub 메시징 기반 비동기 센싱 데이터 수집

엔터프라이즈 환경에서 메시징 시스템은 애플리케이션을 구성하는 소프트웨어 간 데이터를 교환하기 위한 메시지 버스(message bus)인 메시징 기반 미들웨어(Message-Oriented Middleware, MOM)를 사용한다[13]. MOM은 비동기의 데이터 전달을 지원하므로 애플리케이션 수행 과정에서 송신자는 송신된 메시지가 수신자에 의해 수신되거나 처리되기를 대기하지 않는 방식으로 동작한다. 메시지는 독립적인 정보 단위로써 각 메시지는 메시지를 처리하는 엔터프라이즈의 대상 업무가 필요로 하는 자료와 상태를 전달한다.

여기서는 IoT 환경에서 수집된 센서 데이터를 애플리케이션에 전달하기 위해서 파이프-필터(Pipe-and-Filter) 패턴을 이용한다[11]. 파이프-필터 패턴에서 각 컴포넌트는 입력과 출력 집합을 갖는다. 컴포넌트는 입력단(input endpoint)에서 데이터 스트림을 읽고 출력단(output endpoint)에 데이터 스트림을 출력한다. 필터는 데이터 스트림 변환에 사용하며, 변환 결과는 이를 필요로 하는 컴포넌트에 전달한다.

메시징 시스템은 publish-and-subscribe(pub/sub)와 point-to-point queuing(P2P)의 두 가지 유형의 메시징 모델을 제공한다. 파이프-필터를 구현하기에 적합한 pub/sub 모델은 일대다 형태로 메시지를 분배하는 데 활용되며, P2P 모델은 메시지의 일대일 전달을 목적으로 한다. 이러한 메시징 모델의 선택은 애플리케이션의 메시징 요구사항에 따라 달라진다. 특히 메시지가 여러 수신자에게 분배되어야 한다면 메시징 시스템은 pub/sub 모델이 적합하다. 파이프-필터 패턴에서 데이터를 변환하는 컴포넌트가 이를 원격지의 컴포넌트에 전달하기 위해서는 pub/sub 기반의 미들웨어를 활용하도록 한다.

Pub/sub 메시징 모델은 MOM 클라이언트가 메시징 서버(message server) 또는 메시지 브로커(message broker)와 연계하여 메시지를 출판(publish)하고 메시지를 구독(subscribe)하는 방법을 정의한다[8, 9, 10]. Pub/sub 모델에서는 메시지의 출판을 생산(produce)과 구독을 소비(consume)한다는 용어를 사용한다. 대표적인 pub/sub 미들웨어로는 JMS[8]와 NardadaBrokering[9]이 있다.

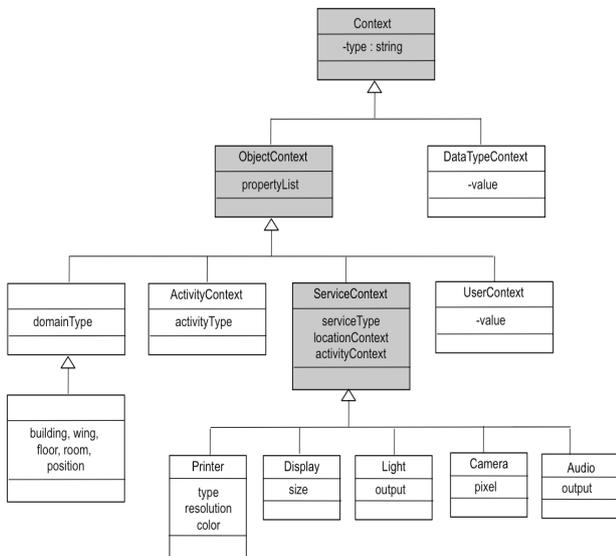


Fig. 4. Classes of context

Pub/sub 모델에서 메시지 생산자는 Topic이라고 불리는 가상 채널을 통해 하나의 메시지를 여러 소비자에게 전송할 수 있다. 메시지 생산자는 메시지를 수신하는 소비자와는 의존성 없이 동작한다. 이는 네트워크 연결이 단절된 상태에서 발생 메시지를 수집할 수 있으므로 단속적인 네트워크 환경에서도 정보 획득이 가능한 장점을 제공한다.

Pub/sub 시스템의 메시지는 응용에 따라 다수의 메시지로 합성될 수 있으며 합성메시지를 통지받을 수 있다. Fig. 5는 이벤트 기반 pub/sub 기반 메시징 프레임워크의 작업 흐름을 보인 것으로 애플리케이션은 생산자로서 메시지를 출판하고 소비자로서 관심 메시지를 구독하며, 소비자의 관심 메시지가 수신되면 메시지를 통지한다. 애플리케이션은 상위 수준의 메시지인 합성 이벤트(composite event, CE)의 수신을 CE 검출 프레임워크에 구독 요청할 수 있으며 CE 검출 프레임워크는 합성 이벤트의 해석 후 구독/통지를 수행하여 메시지를 종합한다[10].

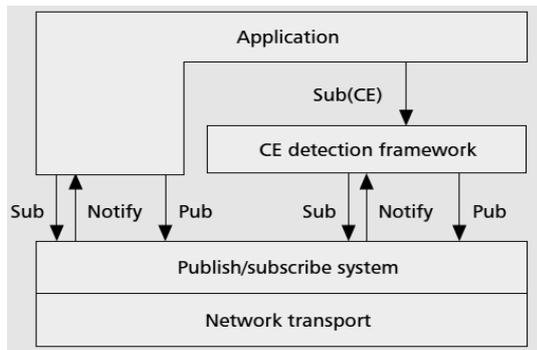


Fig. 5. The overall event flow

### 3.3 IoT 환경의 센싱 정보 수집 체계 구성

IoT 시스템은 사물(Thing), 수집기(Aggregator), 애플리케이션(Application)의 3개 계층(tier)으로 구성된다. 수집기와 애플리케이션과 애플리케이션 간의 연계는 파이프-필터 패턴이 적용된다. Fig. 6은 각 계층에서의 측정정보의 처리를 위한 파이프라인의 상호작용을 보인 것이다. 사물은 에이전트를 통해 센서로부터의 원시 센싱 데이터를 센싱으로부터 획득한 후 메시지로 구성한다. 이는 상위 수준의 메시지인 합성 이벤트로 표현된다.

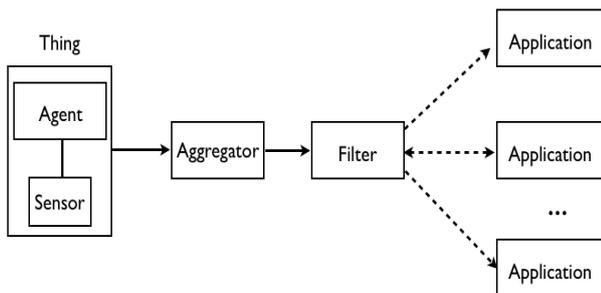


Fig. 6. Pipeline in IoT system

에이전트는 자체 저장장치에 획득 정보를 저장하지 않고 유무선의 네트워크를 통해 수집기에 전달한다. 이때 획득정보 및 알람 정보가 애플리케이션에 필터를 통해 전달된다. 수집기는 다수의 에이전트로부터 데이터를 수집한 후 데이터에 메타데이터를 추가하여 저장한다. 수집기는 주기적으로 수집 자료에 대한 필터링을 수행하여 자료를 가공한다. 이를 위해 필터는 메시지 버스의 역할을 수행하는 비동기 기반 메시징 하부구조인 NaradaBrokering을 사용한다[10].

수집기와 애플리케이션 사이는 필터를 통해 생성자와 소비자 관계(producer-consumer relationship)가 이루어지며 약결합(loose-coupled)으로 연결되어 높은 확장성을 제공한다. 수집기는 메시지의 생성자로 필터를 통해 메시지를 출판하고 애플리케이션은 메시지의 소비자로서 메시지의 구독자로 동작한다. 애플리케이션에서는 구독 메시지의 해석을 통해 센서의 비정상 상태 및 오류 정보를 전달받는다.

### 3.4 실험 환경 및 결과 평가

이 실험에서는 애플리케이션의 검증을 위해 에이전트 시뮬레이터를 Visual C++로 개발하여 Windows에서 동작하도록 한다. 개발된 에이전트는 상수의 수압, 유량 정보를 생성하는 센서를 갖는다. 수집기는 플랫폼에 독립적으로 운영하도록 Java 언어로 개발하며 커널 2.6.x의 Linux 환경에서 수행하였다.

- JDK 1.6.x
- NaradaBroker 4.2.2
- Tomcat 5.33
- Ubuntu 11.04 (Kernel 2.6.x) Linux

에이전트는 획득된 센서의 측정 정보 및 상태 정보 표현을 위해 XML을 적용하여 확장성을 갖도록 하였으며, 에이전트와 수집기는 TCP/IP 애플리케이션 메시지를 통해 전송과 수신이 이루어진다. 메시지는 애플리케이션의 종류에 따라 필터를 통해 전달된다.

Table 1은 센서로부터 획득된 측정 정보 및 상태 정보의 XML 표현으로 XML 메시지를 구성하는 엘리먼트에 대한 설명이다. 수집기의 메시지 전송모듈은 XML 메시지 해석을 위한 메시지 큐 서비스에 전달하고 메시지 해석 모듈은 메시지 큐에서 이를 읽은 후 메시지 종류별로 각각 처리한다.

Table 1. Description of XML message

Element name	Description
Event-Type	Message Type
Event-ID	Message ID
Event-Value	Message Value
Event-Time	Event Time
Device-ID	Device ID
Device-Type	Device Type
Device-Status	Device Status
Parameters	Sensory Value

Fig. 7은 에이전트가 수집기에 전달하기 위한 XML 메시지의 예를 보인 것으로 에이전트는 장치를 등록하기 위한 초기화 메시지를 구성한다. 구성된 메시지는 Naradabrokering에 메시지를 출판하게 되며 해당 메시지는 등록된 수집기에 제공된다.

```
<Event-Message>
  <Event-Type>INI</Event-Type>
  <Event-ID>
</Event-ID>
  <Event-Value/>
  <Event-Time>
    2014-09-05 20:07:07
  </Event-Time>
  <Device-ID>1000</Device-ID>
  <Device-Type>US004</Device-Type>
  <Device-Status/>
  <Parameters/>
</Event-Message>
```

Fig. 7. XML message example

Fig. 8은 사물의 센서값을 획득하는 에이전트의 GUI 화면을 보인 것으로 센서의 측정값을 얻기 위해 에이전트는 측정에 필요한 형상정보를 수집기에 요청한다. 등록 에이전트와 수집기간의 메시지 요청 및 응답으로 이루어진다. 일례로 에이전트는 측정값이 임계치를 초과하는 경우 전달받은 메시지를 수집기에 전달하고 모니터링 애플리케이션에 경고 메시지를 전달한다.



Fig. 8. GUI of agent

Fig. 9는 NaradBrokering을 통해 가시화 애플리케이션의 입력단에 전달된 메시지 내용으로, 해당 메시지는 필터를 통해 특정주기의 센서의 평균값을 전달받는다.

```
Event-Type:EVT
Event-Time:2014-09-05 09:48:18
Device-Type:null
Number_Of_Param:0
Param AVG:NaN
Event-Type:DAT
Event-Time:2014-09-05 09:48:18
Device-Type:US004
Number_Of_Param:7
Param AVG:24.685713
Event-Type:DAT
Event-Time:22014-09-05 09:48:18
Device-Type:US004
Number_Of_Param:7
Param AVG:25.257143
```

Fig. 9. Message contents

#### 4. 결론

IoT는 센서와 제어 기능을 갖는 사물이 사용자와 상호작용이 원활하도록 연결된 분산 컴퓨팅 환경으로, IoT 시스템은 서비스 및 애플리케이션 간의 자료 교환을 위해 상호운용성 지원이 필수적이다. IoT 애플리케이션의 상호운영성을 개선하기 위해 애플리케이션 간의 높은 확장성 및 약 결합을 갖는 pub/sub 기반 메시징 시스템을 사용한다. 이 논문에서는 IoT 애플리케이션의 상호운용성 및 확장성을 갖는 메시징 시스템을 사용하여 IoT 시스템의 개념 프레임워크를 설계하였다. 설계한 IoT 개념 프레임워크는 IoT 환경의 사물의 모니터링을 위한 IoT 애플리케이션을 운영 프레임워크에 적용하였으며, 운영 검증을 위해 NaradaBrokering 미들웨어를 활용하여 모니터링 시스템을 개발하여 검증하였다.

#### References

- [1] Luigi Atzori, Antonio Iera, and Giacomo Morabito, "The Internet of Things: A survey," *Computer Networks: The International Journal of Computer and Telecommunications Networking*, Vol.54, No.15, pp.2787-2805, 2010.
- [2] Jayavardhana Gubbi, Rajkumar Buyya, Slaven Marusic, and Marimuthu Palaniswami, "Internet of Things(IoT): A vision, architectural elements, and future directions," *Future Generation Computer Systems*, Vol.29, No.7, pp.1645-1660, 2013.
- [3] Sun, Enji, Xingkai Zhang, and Zhongxue Li, "The internet of things(IOT) and cloud computing(CC) based tailings dam monitoring and pre-alarm system in mines," *Safety science*, Vol.50, No.4, pp.811-815, 2012.
- [4] Antonio Iera, Christian Floerkemeier, Jin Mitsugi, and Giacomo Morabito, "Guest editorial: The internet of things," *IEEE Wireless Communications*, Vol.17, No.6, pp.8-9, 2010.

[5] Benoit Christophe, Mathieu Boussard, Monique Lu, Alain Pastor, and Vincent Toubiana, "The web of things vision: Things as a service and interaction patterns," *Bell Labs Technical Journal*, Vol.16, No.1, pp.55-61, 2011.

[6] Felix Xiaozhu Lin, Zhen Wang, Robert LiKamWa, and Lin Zhong, "Reflex: using low-power processors in smartphones without knowing them," *SIGPLAN Not.*, Vol.47, No.4, pp. 13-24, 2012.

[7] Rohit Chaudhri, Waylon Brunette, Mayank Goel, Rita Sodr, Jaylen VanOrden, Michael Falcone, and Gaetano Borriello, "Open data kit sensors: mobile data collection with wired and wireless sensors," in *Proceedings of the 10th international conference on Mobile systems, applications, and services*, Low Wood Bay, Lake District, UK. pp.351-364, 2012.

[8] Sun, Java Message Service, <http://java.sun.com/products/jms/>, 2001.

[9] S. Pallickara, G. Fox, "NaradaBrokering: a distributed middleware framework and architecture for enabling durable peer-to-peer grids," in *Proceedings of the ACM/IFIP/USENIX 2003 International Conference on Middleware*, Brazil, 2003.

[10] Pietzuch, P. R., Shand, and B., Bacon, J., "Composite Event Detection as a Generic Middleware Extension," *IEEE Network*, Vol.18, pp.44-55, 2004.

[11] François, A.R.J., *Software Architecture for Computer Vision: Beyond Pipes and Filters*, Prentice Hall, 2003.



### 강 윤 희

e-mail : yhkang@bu.ac.kr

1989년 동국대학교 컴퓨터공학과(공학사)

1991년 동국대학교 컴퓨터공학과(공학석사)

2002년 고려대학교 컴퓨터과학과(이학박사)

2000년~현 재 백석대학교 정보통신학부  
부교수

관심분야: 클라우드 컴퓨팅, 그리드 컴퓨팅, 결합포용



### 고 완 기

e-mail : kwk@chu.ac.kr

1989년 동국대학교 전자계산학과(이학사)

1994년 조선대학교 전자계산학과(공학석사)

2004년 제주대학교 경영학과(박사수료)

1997년~현 재 제주한라대학교 방송영상과  
부교수

관심분야: 데이터베이스, 멀티미디어, 디지털영상처리