

사물인터넷 환경 구축을 위한 응용 프레임워크 연구 동향

1. 서론

인터넷을 통해 유통되는 정보는 인간이 생산한 자료의 처리과정으로 생성되었다. 그러나 인터넷 환경은 물리적인 사물이 센싱(Sensing)과 액추에이션(Actuation) 기능, 데이터 처리, 통신 기능을 담당하며 사이버 공간인 인터넷에 연결됨으로써 모든 사물이 정보의 생산자와 소비자 역할을 함께 할 수 있도록 진보하고 있다. 사물인터넷(IoT, Internet of Things)은 사물이 인터넷에 연결되어 정보를 공유하는 것으로 정의되며, 다양한 센서의 보급으로 IoT 환경이 빠르게 구축되고 있다. Cisco는 2015년에는 2백 50억개, 2020년까지 5백억 개의 센서를 포함한 장치가 인터넷에 연결될 것으로 예상하였다. 이는 데이터 수집이 필요한 센서의 수와 종류가 급속히 증가함을 의미하며, 센싱 데이터 수집 시스템이 사물과 연관된 센서로부터 수집된 정보를 가공 및 처리하기 위해 필요하다.

사물인터넷 (IoT, Internet of Things)은 사물이 인터넷에 연결되어 정보를 공유하는 것으로 정의되며, 다양한 센서의 보급으로 IoT 환경이 빠르게 구축되고 있다.

IoT 환경의 자료처리는 빅데이터의 세가지 특성인 데이터 다양성 (variety), 데이터 크기(volume) 및 데이터 증가 속도(velocity)를 갖는다. IoT 생태계는 각종 센서 및 측정장치에서 수집된 데이터의 빅데이터 처리를 요구한다. Apache 그룹의 Hadoop은 오픈소스를 기반으로 하는 빅데이터 활용을 위한 주요 기술로서 사실적 표준(De facto standard)으로 자리 잡고 있으며, 실시간 데이터 처리를 위한 Apache 그룹의 Flume과 Storm의 활용이 연구되고 있다.

IoT 시스템의 구성요소는 물리적으로 분산되며, 이러한 시스템의 효



강운희
백석대학교 정보통신학부
부교수



올적인 관리를 위해서는 시스템 설계 과정에서 응용 간의 자료교환을 위해 확장성(scalability)과 상호 운영성(interoperability)과 같은 비기능성 요구사항(non-functional requirement)을 고려하여야 한다. 이러한 요구사항을 만족하기 위해 사용자에 IoT 환경에서 다양한 유무선의 네트워크와 연동을 통해 외부에 위치하고 있는 다양한 센서 정보를 수집 또는 제어하여 응용에 정보를 제공할 수 있는 공통 플랫폼(common platform)이 필요하다. IoT 공통 플랫폼의 설계에서는 사물들이 응용과의 슬기없는(seamless) 인터페이스를 지원하기 위한 아키텍처가 요구된다.

본 고에서는 센싱 데이터 수집과 수집된 데이터의 응용 처리를 위한 기반기술 및 참조 아키텍처 설계를 소개한다. IoT 환경의 특성상 다양한 분야의 사용자 유스케이스(use-case)의 요구사항을 지원하기 위한 단일 아키텍처를 구성하는 것이 적합하지 않으므로 여기서 설계된 아키텍처에서는 스트림 기반의 분산 처리를 위한 데이터 활용을 목적으로 한다. 이를 위해 설계된 아키텍처는 메시징 미들웨어를 활용하여 IoT에서 요구되는 사건(Event), 메시지(Message) 및 응용(Application) 간의 확장성 및 상호운용성 문제를 해결하는 것을 목적으로 한다.

본 고의 2장에서는 IoT의 등장 배경, 운영환경의 특징 및 관련연구를 기술한다. 기술되는 IoT 관련연구의 주요내용으로는 오픈소스 기반의 데이터 수집체계로서 Apache Flume과 Apache Storm을 포함하며, 웹 기반 통합을 위한 접근인 WoT(Web of Things)와 모바일 환경을 위한 프레임워크인 ODK를 기술한다. 3장은 센싱 데이터 수집을 위한 IoT 아키텍처 설계를 기술한다. 마지막으로 4장에는 결론을 기술한다.

2. 관련연구

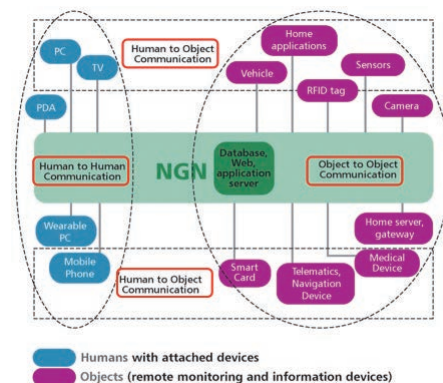
2.1 IoT 기술개요

국제표준화 기관인 EPCglobal의 연구 그룹 Auto-ID

Lab은 RFID(Radio Frequency Identification)의 유일한 식별자인 EPC(Electronic Product Code)를 갖는 사물을 중심으로 초기의 IoT를 정의하였다. 이는 사물에 EPC로 식별 가능한 RFID 태그를 부착하여 전세계에 설치된 RFID 리더를 통해 실시간으로 코드를 읽고 그 정보를 분산시스템에 저장관리 함으로써, 사물의 인식과 글로벌한 위치 추적 및 트래킹을 지원 하였다. 이는 사물의 정보를 실시간으로 모니터링하여 관리를 할 수 있으며, 표준화된 인터페이스를 통해 다양한 IoT 서비스를 가능하게 하였다. 현재 IoT는 기존 인터넷의 개념을 확장하여 다양한 주변 사물이 인터넷에 참여하는 사물 대 사물, 사람 대 사물간의 네트워크를 포괄하는 차세대 인터넷의 패러다임으로 발전하였다. <그림 1>은 사물과 인간이 연결되어진 차세대 네트워크 기반의 인터넷 환경을 보인 것이다.

현재 IoT는 기존 인터넷의 개념을 확장하여 다양한 주변 사물이 인터넷에 참여하는 사물 대 사물, 사람 대 사물간의 네트워크를 포괄하는 차세대 인터넷의 패러다임으로 발전하였다.

IoT 환경은 무선 센서 네트워크(Wireless Sensor Network)와 RFID의 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.



<그림 1> 인간과 사물이 연결되어진 IoT 구성

개인화를 위한 다양한 센서 용·복합 서비스는 센서 자원의 개방화와 센서 네트워크 간의 상호운용성 확보를 위한 기술을 기반으로 하고 있으며, IoT 프레임워크에 대한 설계는 분산 컴퓨팅의 핵심 기술 응용서비스 개발에 필수적이다. 특히, 이종 센서 및 자원들에 대한 상호운용성을 제공하고, 다양한 센싱 정보에 대한 동일한 의미부여 및 해석, 다계층 사용자에 대한 체계적인 서비스 개발, 배포, 운영 등을 가능하게 하는 표준적인 정보 처리 체계가 요구된다. 의미해석은 네트워크 컴퓨팅 기술과 더불어 스마트 센서가 시공을 초월해 필요한 정보를 얻고 각종 장치들이 스스로 기능을 수행하는 데 있어 중추적인 역할을 할 수 있기 때문이다. 이러한 센서의 지능화는 전통적인 센서 활용 분야를 뛰어넘어 스마트 홈 시스템, 원격진료 시스템, 스마트 빌딩, 지능형 자동차, 헬스케어 및 동식물까지 연결된 초연결 사회 구축 및 미래 신시장 창출을 위해 센서의 활용 영역을 넓히고 있다.

가트너는 2012년 IoT의 필수 기술로서 저전력 네트워크, 센서 데이터 최적화 및 관리기술, 저전력 임베디드 OS 기술, 신규 전력공급 및 저장기술과 저비용/저전력 프로세서기술을 정의하였다. 이러한 사물인터넷의 주요 요소는 센싱기술, 유무선 통신 및 네트워크 기술과 서비스 인터페이스(Service Interface) 기술의 세가지로서 소프트웨어 아키텍처로서 조직화 할 수 있다.

센싱 기술은 필요한 사물이나 장소에 전자태그(Tag)를 부착해 주변 상황의 정보를 획득하고, 실시간으로 정보를 전달하는 IoT의 핵심 기술이다. 센싱을 위한 IoT 디바이스의 개발과 운영을 위해서는 오픈소스 플랫폼을 활용된다. 현재 IoT 환경에서의 디바이스는 다음의 세가지로 분류되어 사용되고 있다.

- 임베디드 8비트 SoC(System-On-Chip) 제어 장치는 저렴한 오픈소스 하드웨어 플랫폼인 아두이노(Arduino)를 활용한다. 아두이노는 다중 플랫폼으로 Microsoft Windows, Macintosh OSX 및 Linux 상에서 동작되며, 개발이 편리한 프로그래밍

사물인터넷의 주요 기술요소는 센싱기술, 유무선 통신 및 네트워크 기술과 서비스 인터페이스(Service Interface) 기술의 세가지 기술요소로서 소프트웨어 아키텍처로서 조직화 할 수 있다.

환경을 제공한다. 특히, 아두이노는 소프트웨어와 하드웨어 모두 오픈소스로서 사용자의 목적에 따라 확장 가능하다. <그림 2>는 아두이노 우노 보드를 보인 것이다.

- Atheros와 ARM을 기반으로 한 시스템은 제한적인 32비트 아키텍처로서 소형 가정 라우터 및 다양한 장치개발에 활용할 수 있다. 임베디드 리눅스 플랫폼인 OpenWRT 또는 전용 임베디드 운영체제를 사용하고 있으며 최근에는 아두이노 Zero 또는 아두이노 Yun이 플랫폼으로 등장하고 있다.
- 라즈베리파이를 활용한 32비트 또는 64비트 아키텍처의 시스템은 신용카드 크기의 디바이스로서 모니터와 키보드 마우스를 연결하여 PC로 사용 가능하며, 비디오 출력을 위한 RCA(Radio Corporation of America)와 HDMI(High Definition Multimedia Interface) 포트 및 통신을 위한 LAN 포트를 포함하고 있다. 다수의 GPIO와 저장장치로서 SD 카드를 갖는다. <그림 3>은 라즈베리파이 기반 디바이스를 보인 것이다.

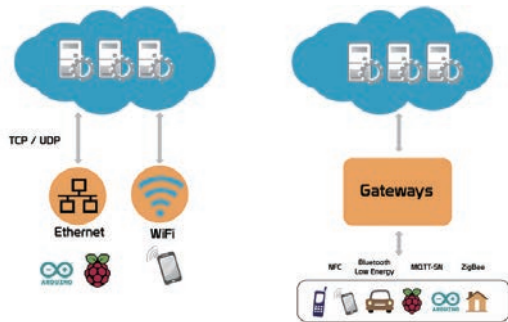
디바이스와 인터넷간의 통신은 직접적인 방식과 계



<그림 2> 아두이노 우노(UNO 보드)



<그림 3> 라즈베리 파이 기반 디바이스

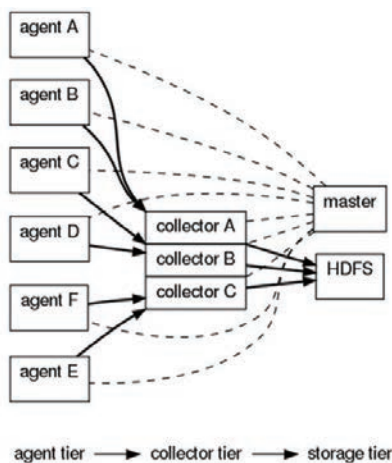


〈그림 4〉 장치 인터넷 통신 모드

이트웨이를 활용한 통신 방식의 두 가지 모드가 사용될 수 있다. 〈그림 4〉는 디바이스와 인터넷 통신 모드를 보인 것이다. 이를 위해 사용되는 유무선 통신 및 네트워크 기술은 사물이 인터넷에 연결 되도록 지원하는 기술로 NFC(Near Field Communication), URRT ZigBee, Bluetooth Low Energy, WiFi를 포함한 유무선 통신망으로 구성된 네트워크 환경에서 연결성 보장 및 효율적인 망관리 기술을 의미한다.

2.2 Apache Flume

Apache Flume은 대량의 데이터를 안정적이고 효율적으로 스트림 기반(stream based)으로 전달할 수 있는 오픈소스 기반의 시스템이다. IoT 환경에서 생성된 빅데이터의 일괄처리(batch processing)에 적합한 Hadoop



〈그림 5〉 Flume 기반 데이터 수집 프로세스

의 활용을 위해서는 먼저 Hadoop 응용에 필요한 자료를 처리하기 위해 해당 자료를 HDFS에 저장하여야 한다. HDFS에 파일 저장은 수집된 자료를 프로그램을 통해 파일 단위로 저장할 수 있지만, 관리 측면에서 서버 수의 증가에 따라 생성된 로그 파일을 지속적으로 HDFS로 옮기기란 쉽지 않다. 이러한 이유로 Flume은 로깅 시스템에 사용되며, 다양한 장비에서 생성된 데이터를 Hadoop의 저장소인 HDFS에 저장할 수 있게 데이터를 이동시키는 역할을 한다. Flume은 마스터(master)와 노드(node) 두 가지 기능으로 구분된다. IoT 환경에서 Flume은 분산 구조로 데이터양이 급증해도 기존 서버의 설정이나 구

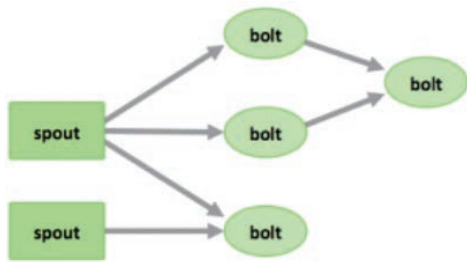
Apache Flume은 대량의 데이터를 안정적이고 효율적으로 스트림 기반(stream based)으로 전달할 수 있는 오픈소스 기반의 시스템이다.

조 변경 없이 손쉽게 확장할 수 있는 이점이 있다. Flume을 활용한 센서데이터 수집은 3-tier 구조로서 IoT 환경의 센서로부터 생성된 자료를 대행하는 에이전트(agent)로부터의 센싱 데이터를 콜렉터(collector)를 통해 HDFS에 단일방향으로 저장할 수 있다. HDFS에 저장된 로그 파일을 분석 및 처리하기 위해서는 MapReduce 프로그램이 사용한다. 〈그림 5〉는 Flume을 사용한 데이터 수집 프로세스를 보인 것이다.

2.3 Apache Storm

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.

IoT 환경에서 센서 도입이 빠르게 이루어짐에 따라 센서로부터 생산되는 스트리밍 기반 자료의 가공은 효율적인 빅데이터 처리 및 분석이 필수적이다. 스트리밍 기반의 실시간 데이터 처리를 위한 분산 프레임워크인 Apache Storm은 2011년 트위터에 의해 인수한 기업인 BackType에서 시작한 프로젝트로서 트위터에서 사용하



〈그림 6〉 스파우트와 볼트로 구성된 토폴로지

는 실시간 분석 시스템이다. 2014년에는 Apache의 최상위 오픈소스 프로젝트로 성장하였다.

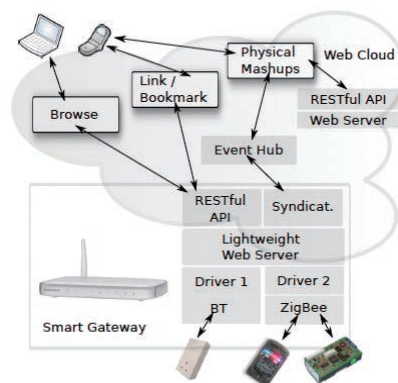
Apache Storm 아키텍처는 Hadoop에서 배치형식으로 MapReduce 작업을 실행하는 반면 Storm에서는 작업종료 없이 토폴로지(Topology) 작업을 수행하는 차이점을 갖는다. MapReduce 작업과 토폴로지의 차이는, MapReduce 작업은 정해진 데이터 세트를 처리한 후 완료되지만 토폴로지는 계속해서 메시지를 처리한다는 점이다. 이러한 데이터 처리는 배치형식의 빅데이터 처리의 제약점을 해결하고 IoT 환경에서 센서로부터 수집된 실시간 데이터 분석을 가능하게 한다. 전통적인 실시간 데이터 처리에는 복잡한 메시지 큐를 이용해 메시지 스트림을 처리가 사용되는데 개발자가 메시지 흐름의 복잡도를 고려해야 하는 어려움이 있다. Storm은 DAG(Direct Acyclic Graph)로 표현되는 토폴로지를 정의하여 데이터의 소스, 처리와 출력모듈을 정의할 수 있는 기능을 통해 개발자의 복잡한 메시지 처리의 어려움을 해결한다. 이를 위해 Storm은 데이터 소스인 입력스트림을 스파우트(Spout)로 처리와 출력모듈을 볼트(Bolt)로 정의한다. 스파우트는 스트림의 소스를 지칭하는 컴포넌트이다. 메시지는 튜플(Tuple)로 표현한다. 일반적으로 스파우트는 외부 소스로부터 튜플을 읽고 토폴로지로 튜플을 생성해 스트림을 생성하는 역할을 한다. Storm은 스트림을 처리하는 도중 실패했을 때 결과를 재전송하는 메시지 전송의 신뢰성을 보장한다. 스파우트와 볼트를 통해 전달되는 자료는 하나 이상의 스트림에 튜플을 전송할 수 있다. 〈그림 6〉에서와 같이 토폴로지가 시작되면, 스파우트는 입력 소스로부터 데이터를 가져온 후, 데이터 변환 또는 필터 작업을 수행하는 볼트로 데이터를 전달한다. IoT 환

경에서 Storm은 모든 형태의 소스로부터 데이터를 수신할 수 있는 스파우트 개발의 용이한 점과 기존에 풍부한 스파우트가 있다. 트위터 스트리밍 API, Apache 카프라(Kafka), JMS 브로커 등 다양한 소스에서 기존에 개발된 스파우트의 재사용이 가능하다. 〈그림 6〉은 스파우트와 볼트로 구성된 토폴로지를 보인 것이다.

2.4 WoT

IT 기술의 발전과 개방된 개발환경에 대한 요구가 증가됨에 따라 웹과 같이 쉽고 단순한 응용개발 환경에 대한 관심이 증가하고 있다. IoT 환경은 물리적인 장치와 응용 연계에 필요한 개발 시스템 간의 상호 호환이 체계적인 통합 시스템 구축을 위한 필수적인 사항이다. 웹은 사용자 측면에서 사물에 관련된 장치를 접근하고 동작시키기 위해 하위 네트워크에 상관없이 표준 HTTP 프로토콜을 사용할 수 있도록 한다. 이는 실제 사물이 웹 환경으로 통합되고 이들이 웹을 통해 접근 가능한 하나의 서비스로서 다루어질 수 있다.

WoT(Web of Things)는 서로 연결된 모든 종류의 디바이스에서 접근 가능한 분산된 응용과 서비스로, 이 모든 기반에서 웹은 인터넷을 통해 정보자원의 접근이 용이한 환경으로서 IoT의 사물에 대한 접근성을 높이기 위해 모든 사물을 웹으로 통일하기 위해 제안하였다. WoT 관련하여 표준개발이 ITU-T에서 이루어지고 있으며 ITU-T SG14에서 WoT의 프레임워크인 Y.WoT가 개발 진행 중이다.



〈그림 7〉 WoT 기반의 장치 통합

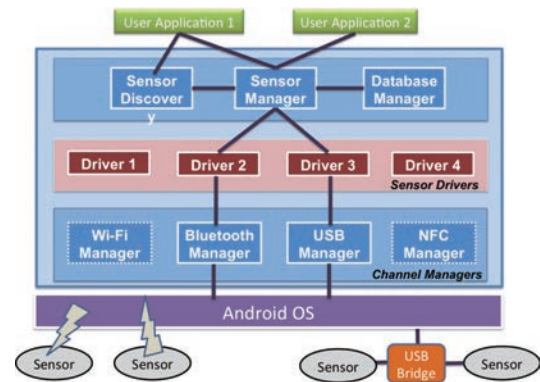


WoT에서는 센서, 장치와 액츄에이터들을 웹의 주소체계를 기반으로 하는 객체로 표현하고 스마트 게이트웨이를 통하여 이를 등록하도록 함으로서 궁극적으로 물리적 사물을 웹 인터페이스를 통해 검색하고 제어하고 조합한다. 이들 간의 통신은 REST를 기반으로 하며, 물리적 매쉬업(mash-up)까지 지원한다. 개념적으로 WoT는 기존의 객체 식별을 위해 사용하고 있는 RFID의 확장으로서 인터넷 기반 연계를 위한 주소체계를 구성한다. WoT는 웹을 사용하여 사물들이 통신할 수 있도록 지원하는 것을 목적으로 한다. <그림 7>은 WoT 기반의 장치 통합을 보인 것으로 응용의 관점에서 각각의 사물은 단지 웹의 자료 또는 서비스로 보이며, 접근 및 활용이 가능하게 된다. 디바이스는 웹서버를 구동시킨 후 웹을 통해 접근 할 수 있도록 함으로서 웹 클라이언트에서 디바이스에 별도의 특정 프로토콜을 사용하지 않고 웹 프로토콜만을 이용할 수 있다.

2.5 ODK

센서 및 스마트 디바이스 모두가 인터넷 연결을 기반으로 정보를 교환하는 WoT와 달리 IoT 환경에서 디바이스 간 통합은 스마트 폰을 포함한 디바이스와 센서에 내장된 네트워크 인터페이스를 이용해서 두 기종 간 직접적인 방식을 통한 이루어진다. 이는 다른 네트워크 환경에 영향 받지 않고 독립적이고 개인화된 네트워크를 구축할 수 있는 장점을 갖으며, 생활 속의 다양한 스마트 디바이스는 다양한 네트워킹 방식을 지원하는 소규모 네트워크 방식이 필요하다.

ODK(Open Data Kit) 센서 프레임워크는 워싱턴 대학교(University of Washington)이 개발하였으며 센서 관리 계층(sensor management) 계층을 두어 외부로부터 모바일 환경에서 외부 센서로부터 수집된 정보를 스마트 기기에서 활용할수록 지원한다. <그림 8>에서와 같이 외부 센서의 정보를 수집 및 관리하여 이를 응용계층에 전달하고 네트워킹 계층까지 추상화함으로써 센싱 데이터의 수



<그림 8> ODK 프레임워크

집 및 모니터링 응용을 위해 센서 통합과정을 단순화하기 위해 설계할 수 있도록 지원한다. ODK는 한 개의 외부 센서로부터 전달된 정보를 다수의 응용 서비스가 공동으로 활용할 수 있게 함으로서 효율적인 자원 관리 환경을 제공하며, 센서 정보에 대한 통합 보안 관리를 가능하도록 함으로서 효율적으로 리소스 관리가 가능하다.

3. 센싱 데이터 수집 IoT 아키텍처

3.1 IoT 서비스 플랫폼 요구기능

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
 - 인터넷에 연결된 사물이나 기기에 접근하기 위한 플랫폼 등록을 수행
 - 등록 기기에 대한 검색, 추가, 삭제, 수정 요청을 지원함
- IoT 사용자 프로파일 관리
 - 사용자의 인증 및 접근 제어에 위한 정보의 등록을 수행
 - 사용자 및 사용자 그룹에 대한 서비스 권한을 관리함
 - 사용자 클라이언트인 웹 및 스마트 기기의 앱을 통한 접근 서비스 지원
- IoT 서비스 및 응용 프로파일 관리
 - 센서를 갖는 사물로 부터의 데이터 수집, 저장 및 처리를 위한 응용 및 서비스 정보 관리
 - 모든 데이터 처리는 인터넷을 통해서 이루어지며 클라우드 기반의 플랫폼 운영을 지원함

IoT 서비스 플랫폼은 다수의 센서로부터 생성된 자료의 처리를 위한 확장 가능한 수집체계를 필요로 한다. 스트리밍 기반의 데이터 처리 환경에서 메시징 시스템(messaging system)은 응용에 필요한 데이터 전달을 효율적으로 지원함으로써 응용 수행 과정에서 송신자(sender)는 송신된 메시지가 수신자(receiver)에 의해 수신되거나 처리되기를 대기하지 않는 방식으로 동작한다. 메시지는 독립적인 정보단위로서 각 메시지는 메시지를 처리하는 응용의 대상 업무가 필요로 하는 자료와 상태를 전달한다.

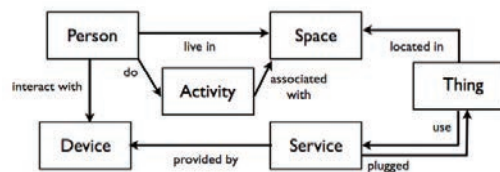
3.2 IoT 개념 프레임워크

IoT 환경은 사물과 사람의 상호작용이 이루어지며, 이 과정에서 다양한 서비스를 지원받게 된다. 다양한 서비스와 콘텐츠는 상호연동을 통해 운영되며, 사용자의 디바이스와 사물은 서비스로 구성된 응용 수행을 위해 메시징

서비스를 통해 정보를 교환한다. <그림 9>는 설계된 IoT 개념 프레임워크를 보인 것이다.

설계된 IoT 개념 프레임워크에서 IoT 환경은 다수의 Space에 구성되며, 단위 Space는 IoT의 컨텍스트(context) 구조를 운영한다. Person은 Activity에 참여하며, Space에서 여러 Activity를 수행한다. Thing은 센서와 관계하며, Space에서 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 응용에서 공통적으로 필요로 하는 소프트웨어의 부분이다. 응용은 특정한 도메인을 위해서만 활용되는 큰 범위의 소프트웨어라고 할 수 있다.

IoT 프레임워크에서는 운영환경에서 센서에 의한 통지 또는 사람에 의한 통지 등 시스템 외부에서는 환경 변화에 의해 메시지를 발생한다. 메시지는 센서의 원시 데이터(primitive data)와 구분되고 센서로부터 수집되어진 세부사항을 은닉할 수 있도록 한다. 원시 데이터는 센서의 유형에 따라 특별한 자료 형식과 표현 값을 표현하며 메시지는 원시데이터를 추상된 값으로 조정된다. 메시지는 다수의 메시지의 합성을 통해 새로운 메시지로 구성한다. 이벤트(Event)는 모니터링이 필요한 사건으로서 수신된 메시지의 해석으로서 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.



<그림 9> 설계된 IoT 개념 프레임워크



3.3 IoT 센싱정보 수집 체계 구성

IoT 환경과 같이 사물과 사람의 상호작용이 이루어지며, 이 과정에서 스트리밍 기반으로 수집되어진 데이터의 수집을 위한 체계는 다양한 서비스를 지원하기 위해서는 설계가 필요하다. 이러한 스트리밍 기반 데이터 처리는 다양한 서비스와 콘텐츠는 상호연동을 통해 운영되며, 사용자의 디바이스와 사물은 서비스로 구성된 응용 수행을 위해 메시징 서비스를 통해 정보를 교환한다.

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 응용의 메시징 요구 사항에 따라 달라진다. 특히 메시지가 여러 수신자에게 분배되어야 한다면 메시징 시스템은 pub/sub 모델이 적합하다. 파이프-필터 패턴에서 데이터를 변환하는 컴포넌트가 이를 원격지에 컴포넌트에 전달하기 위해서는 pub/sub기반의 미들웨어를 활용하도록 한다. Pub/sub 모델에서 메시지 생산자는 Topic 이라고 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.

설계된 IoT 센싱정보 수집 체계는 사물(Thing), 수집기(Aggregator), 응용(Application)의 3개 계층(tier)으로 구성된다. 수집기와 응용과 응용 간의 연계는 파이프-필터 패턴이 적용된다. 사물은 에이전트를 통해 센서로부터의 원시 센싱 데이터를 센싱로부터 획득한 후 메시지로 구성한다. 이는 상위 수준의 메시지인 합성 이벤트로 표현된다. 에이전트는 자체 저장장치에 획득 정보를 저장하지 않고 유무선의 네트워크를 통해 수집기에 전달한다. 이때 획득정보 및 알람 정보가 응용에 필터를 통해 전

달된다. 수집기는 다수의 에이전트로부터 데이터를 수집한 후 데이터에 메타데이터를 추가하여 저장한다. 수집기는 주기적으로 수집자료에 대한 필터링을 수행하여 자료를 가공한다. 이를 위해 필터는 메시지 버스의 역할을 수행하는 비동기 기반 메시징 하부구조를 사용한다.

수집기는 사물로부터 데이터를 수집한 후 데이터에 메타데이터를 추가하여 저장한다. 수집기는 주기적으로 수집자료에 대한 필터링을 수행하여 자료를 가공한다. 이를 위해 필터는 메시지 버스의 역할을 수행하는 Apache Flume 또는 Storm 을 사용한다. 사물은 내부적으로 센서를 가지며 자체 저장장치에 획득 정보를 저장하지 않고 유무선의 네트워크를 통해 수집기에 전달한다. 이때 클라우드 컴퓨팅 자원에서 동작되는 응용은 획득정보 및 알람 정보를 처리하며, 응용에 필요한 정보는 필터를 통해 전달된다. 수집기와 응용 사이에는 필터를 통해 생성자와 소비자 관계(Producer-Consumer Relationship)가 이루어지며 약결합(Loose-Coupled)으로 연결되어 높은 확장성을 제공한다. 추가적으로 응용에서는 구독 메시지의 해석을 통해 센서의 비정상 상태 및 오류 정보를 전달받는다.

설계된 IoT 센싱정보 수집 체계는 사물(Thing), 수집기(Aggregator), 응용(Application)의 3개 계층(tier)으로 구성된다. 수집기와 응용과 응용 간의 연계는 파이프-필터 패턴이 적용된다.

4. 결론

IoT는 센서와 제어기능을 갖는 사물이 사용자와 상호작용이 슬기없이 연결된 분산 컴퓨팅 환경으로 IoT 시스템은 서비스 및 응용간의 자료교환을 위해 상호운용성 지원은 필수적이다. 서비스 인터페이스 기술은 IoT 구성 요소들을 서비스(Service) 및 응용(Application)과 연동하도록 하는 역할을 수행한다. 서비스 인터페이스는 응용 및 서비스의 체계적 관리를 수행하며, 응용 및 서비스 운영을 위한 IoT 플랫폼 설계가 필요하다. IoT 응용 개발에서 서비스 플랫폼은 다양한 사물들과 응용을 연결해주며, 이를 통해 IoT 센서 및 단말과 서비스 개발 비용과 시간을 단축할 수 있다. 체계적인 센싱 데이터 수집 시스템은 사물이 지능화되어 인터넷 상의 사용자에게 주변의 다양



한 정보를 제공함으로써 새로운 서비스 생태계 형성을 용이하게 한다.

본 고에서는 IoT 관련연구의 주요내용으로는 오픈소스 기반의 데이터 수집체계로서 Apache Flume과 Apache Storm을 포함하며, 웹 기반 통합을 위한 접근인 WoT(Web of Things)와 모바일 환경을 위한 프레임워크인 ODK를 살펴보았으며, IoT 응용의 상호운용성 및 확장성을 갖는 메시징 시스템을 사용하여 IoT 시스템의 개념 프레임워크를 설계하였다. IoT 응용의 상호운영성을 개선하기 위해 높은 확장성 및 약 결합을 갖는 메시징 시스템을 사용하였으며, 설계한 IoT 개념 프레임워크는 IoT 환경의 사물의 모니터링을 위한 IoT 응용을 운영 프레임워크에 적용할 예정이다.

참 고 문 헌

- [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 Sodi, 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, 2012, p. 351–364.
- [8] Sun, Java Message Service, <http://java.sun.com/products/jms/>, 2001.
- [9] S. Pallickara and 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] Francois, A.R.J., *Software Architecture for Computer Vision: Beyond Pipes and Filters*, Prentice Hall, 2003.



강윤희

- 1989년 동국대학교 컴퓨터공학과 (공학사)
- 1991년 동국대학교 컴퓨터공학과 (공학석사)
- 2002년 고려대학교 컴퓨터과학과 (이학박사)
- 2000년~현재 백석대학교 정보통신학부 부교수

〈관심분야〉

클라우드컴퓨팅, 그리드 컴퓨팅, 결합포용