

---

# Machine Socialization 스키마를 활용한 기기 관리 시스템

김용준 · 유상근 · 정영식 · 김용운 · 정희경

배재대학교 컴퓨터공학과

Device Management Systems

using the schema of the Machine Socialization

Wung-Jun Kim · Sang-keun Yoo · Young-Sik Jung · Yong-Un Kim · Hoe-Kyung Jung

Department of Computer Engineering, PaiChai University

E-mail : y199073@naver.com, lobbi@etri.re.kr, jys@etri.re.kr, qkim@etri.re.kr, hkjung@pcu.ac.kr

## 요 약

최근, 전 세계 주요 국가들의 전략사업으로 지정된 IoT(Internet of Things)의 발전으로 인해 사물끼리의 통신이 가능하게 되었다. 하지만 기기의 정보들을 표현하는 표준이 정해지지 않아 서로 다른 플랫폼을 가진 기기들 간의 정보전달이 어려운 실정이다.

본 논문에서는 기존의 사람들이 직접 기기에 명령하는 방식과 단순 센서를 이용한 기기 제어방식과는 달리 Socialization 표현 언어인 XML 스키마를 활용하여 기기의 정보, 작업의 정보, 기기와 기기의 관계 정보를 나타낸 Machine Socialization 스키마를 제안하고 이를 사용하여 기기를 관리하는 기기 관리 시스템을 제안하였다.

## ABSTRACT

In recent years, communication between objects is made possible due to the development of IoT(Internet of Things) to the specified business strategy of major countries around the world. However, the standard for representing information of the device is difficult not been set the communication between the devices with different platforms.

This paper shows the Conventional people instrument using the method and the simple sensor to the command directly to the machine control system and the information of the devices utilizing different Socialization expression language, XML Schema, information on the job, relationship information of the equipment and devices Machine Socialization proposes a scheme and use them to propose a device management system for managing devices.

## 키워드

스키마, Device Management, IoT, Machine Socialization

## I. 서 론

최근 주요 국가들 사이에서 전략사업으로 대두되고있는 IoT는 가전제품, 전자기기뿐만 아니라 헬스케어, 스마트홈, 스마트카 등 다양한 분야에서 사물을 네트워크로 연결해 정보를 공유할 수 있는 사물네트워크이다[1, 2]. IoT의 발전으로 다

양한 방법을 통해 사물과 사물의 연결이 가능하게 되었다. 현재 국내 IoT시장에선 외부에서 가정의 기기들의 상태를 체크하고 제어할 수 있는 시스템이 상용화 되고 있다[3]. 하지만, 이러한 시스템들은 자사에서 만든 플랫폼만을 개발하여 사용하기 때문에 타사 기기와는 같이 사용할 수 없다는 문제점이 있다. 이러한 문제로 전 세계 표준연

구단체에서 IoT 표준화 작업을 진행하고 있다.

이에, 본 논문에서는 서로 다른 플랫폼을 탑재한 기기들 사이의 원활한 소통을 가능하게 하는 Machine Socialization 스키마를 정의하고 스키마를 활용한 기기 관리 시스템을 제안한다[4].

## II. 관련 연구

### 2.1 XML스키마

XML 스키마는 XML 객체의 속성과 요소 간 상호 관계의 추상적 표현으로 문서에서 스키마를 나타내기 위해 구조 분석과 각 구조 요소를 정의한다. 문서 형 정의나 단순 객체 XML과 같은 언어보다 다소 유리하고 자기 기록, 자동 스키마 생성, XSL 변환을 통한 정의 능력 등의 이점이 있다. 이러한 스키마의 이점을 바탕으로 본 논문에서는 기기 정보 스키마, 작업 정보 스키마, 관계 정보 스키마를 정의하였다.

그림 1은 본 시스템에서 기기 정보를 표현한 기기 정보 스키마이다.

```

<DEVICE_INFO>
  <DEVICE_ID>01</DEVICE_ID>
  <DEVICE_DESCRIPTION>WINDOW</DEVICE_DESCRIPTION>
  <MODEL>SHV-E210</MODEL>
  <DEVICE_NETWORK>LGU+</DEVICE_NETWORK>
  <DEVICE_BOARD>smdk4x12</DEVICE_BOARD>
  <DEVICE_BRAND>samsung</DEVICE_BRAND>
  <WORK_TIME>2</WORK_TIME>
  <DEVICE_PLATFORM>ANDROID</DEVICE_PLATFORM>
  <VERSION>4.4.4</VERSION>
  <SENSOR>
    <SENSOR_NAME/>
    <SENSOR_FUNC/>
    <VALUE/>
  </SENSOR>
  <DEVICE_FUNCTION>
    <FUNC_ID>5001</FUNC_ID>
    <FUNC_DESC>WINDOW_OPEN</FUNC_DESC>
    <ORDER>1</ORDER>
    <FUNC_NAME>OPEN</FUNC_NAME>
  </DEVICE_FUNCTION>
  <DEVICE_FUNCTION>
    <FUNC_ID>5002</FUNC_ID>
    <FUNC_DESC>WINDOW_OPEN</FUNC_DESC>
    <ORDER>1</ORDER>
    <FUNC_NAME>OPEN</FUNC_NAME>
  </DEVICE_FUNCTION>
  <DEVICE_STATE>
    <STATE>close</STATE>
    <STATE_WORK/>
  </DEVICE_STATE>
</DEVICE_INFO>
    
```

그림 1. Device\_Info Schema

## III. 시스템 설계

본 장에서는 Machine Socialization 스키마를 활용한 기기 관리 시스템 설계에 대해 기술한다.

기기 관리 시스템을 구축하기 위한 여러 방법들이 중 AP에 서버를 두고 IP를 할당하여 서버에 PHP로 구성된 Device Manager를 두는 형식으로 설계하였다.

Machine Socialization 스키마는 디바이스 고유 번호, 기능설명, 작업시간 등 디바이스의 정보를 갖고 있는 Device\_info와 작업 그룹 고유 번호, 작업 그룹 설명, 작업 우선순위 등 작업 정보를

갖고 있는 Task\_info, 디바이스 디바이스의 관계를 정의한 Relation으로 나누어져 있다.

이러게 정의된 스키마를 바탕으로 사용자나 센서, SNS 등에서 발생한 이벤트가 가정 내 AP에 있는 서버로 가면 서버 내의 Device Manager는 이벤트에 맞는 작업을 Task\_info에서 찾아 시나리오를 실행한다. 각각의 시나리오는 우선순위를 갖고 있으며, 여러 작업이 실행 될 때 쓰이게 된다. 실행되는 시나리오는 Device\_info 안에 있는 기기 정보를 바탕으로 작업 순서에 맞추어 정보를 주고 받으며 실행된다.

그림 2는 Device Manager시스템의 구성도를, 그림 3은 시스템 아키텍처를 나타낸다.

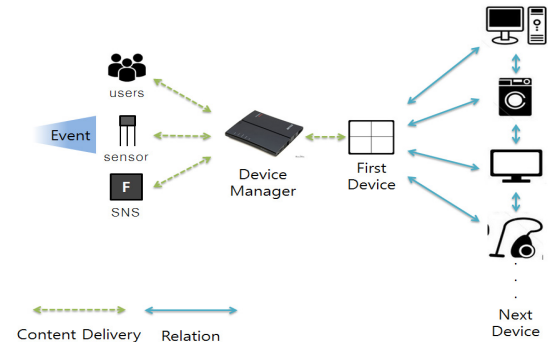


그림 2. Machine Socialization Block Diagram

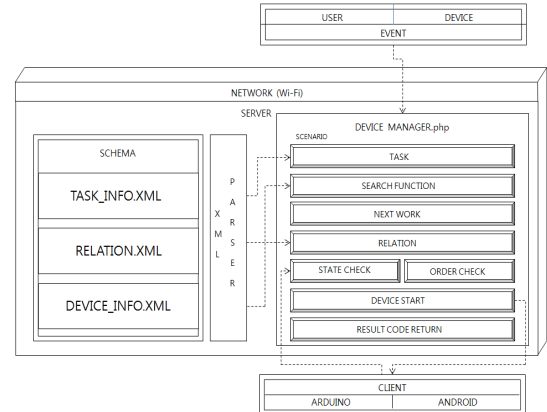


그림 3. Machine Socialization System Architecture

Device Manager는 Event에 맞추어 Task를 선정하고 알맞은 Task 안에 저장되어 있는 1순위 작업기기를 스키마에서 가져와 실행한다. 기기를 실행하기 전에 Device\_info에 있는 기기 정보를 확인하여 기기의 상태와 우선순위를 확인한 뒤 실행한다. 첫 기기의 실행이 완료되면 Device Manager는 동작이 완료된 기기와 다른 기기와의 관계를 찾아 다음 작업에 실행될 기기를 탐색하고 다시 Device\_info에서 기기의 상태와 우선순위를 검사하고 실행시킨다. 이 환경에서 작동되는 Device는 크로스 플랫폼인 Arduino를 사용하여 운영 체제에 제한을 받지 않도록 하였다.

그림 4는 Device Manager의 동작 과정을 나타낸 알고리즘 순서도이다.

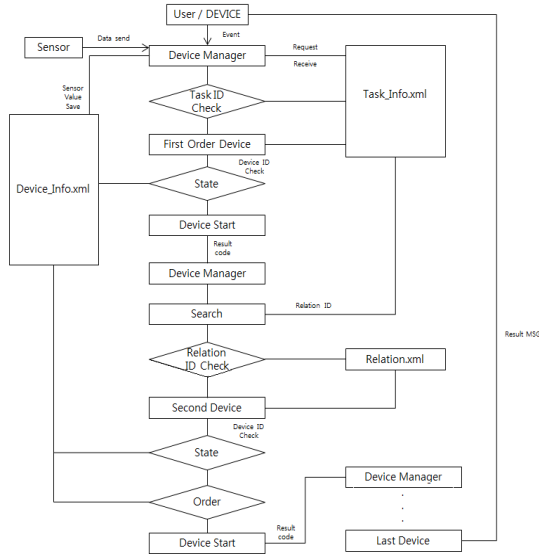


그림 4. Device Manager Algorithm

#### IV. 시스템 구현

본 논문에서의 기기 관리 시스템의 구현환경은 아래 표 1과 같다.

표 1. 개발환경

개발 도구	Eclipse APM_SETUP
개발 언어	JAVA, PHP, Arduino
서버	windows7, Intel i7-2600, 8GB RAM
ARDUINO	UNO, Wi-Fi Shield
디바이스	온도센서, 모터, 청소기, 선풍기
스마트폰	android 4.3 Galaxy S3

청소기와 선풍기에 변압기를 연결하여 Arduino 보드에서 전력을 제어할 수 있도록 하였고, 아크릴로 창문 모형을 만들고 모터를 연결하여 창문이 열고 닫힘을 보이도록 구현하였다. 작업은 온도 조절 작업과, 청소 작업으로 나누었고 2개의 작업이 창문에서 겹쳐지도록 구현하였다. 센서에서 받아진 온도 값은 Device Manager를 통해 Device\_Info에 저장되고 저장된 온도 값을 보고 온도 조절 작업에 대한 Event가 발생한다. 청소 작업은 적정 시간이 되는 실행 되도록 하고 두 작업 간 우선순위는 청소 작업을 더 높게 설정하였다.

각각의 기기들은 실행이 완료되면 Result\_Code를 반환하고 이를 보고 다음 작업으로 진행 된다.

그림 5는 기기 관리 시스템의 시나리오 흐름도를 나타낸다.

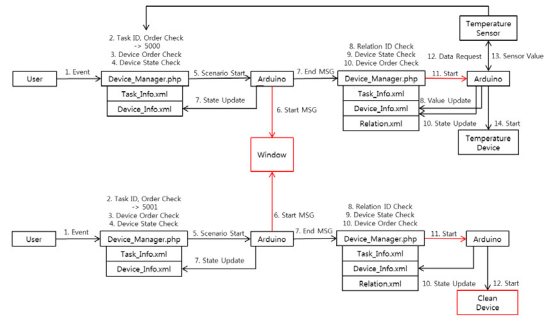


그림 5. Device Manager Task Flow Chart

#### V. 결론

본 논문에서는 Device\_Info, Task\_Info, Relation 스키마와 Device Manager를 AP 내의 서버에 두고 Event에 따라 기기들이 협업하여 실행될 수 있는 Machine Socialization Device Manage 시스템을 제안하였다. 기기와 기기가 서로 관계를 맺고 통신하여 시나리오를 토대로 동작한다.

기존의 회사마다 다른 기기 정보 규격이 있고, 다른 플랫폼을 사용하는 방식과는 다르게 본 논문에서는 Machine Socialization 표준 스키마를 제안하고 디바이스에 Arduino를 사용하여 데이터 전달이 가능하도록 하였다.

향후과제로 기기의 On/Off에 따른 동작 시스템이 아닌 기기의 여러 기능에 따른 동작이 가능하도록 하는 방식의 연구가 필요하다.

#### 감사의 글

본 연구는 미래창조과학부의 지원을 받는 (방송통신표준기술력향상사업 또는 정보통신표준화 및 인증지원사업)의 연구결과로 수행되었습니다.

#### 참고문헌

- [1] 표철식, et al. "IoT (M2M) 기술 동향 및 발전 전망." 한국통신학회지 (정보와통신) 30.8 (2013): 3-10.
- [2] Bandyopadhyay, S., Balamuralidhar, P., & Pal, A. (2013). Interoperation among IoT Standards. Journal of ICT Standardization, 1(2), 253-270.
- [3] Viani, Federico, et al. "Wireless architectures for heterogeneous sensing in smart home applications: Concepts and real implementation." Proceedings of the IEEE 101.11 (2013): 2381-2396.
- [4] Amano, Shun'ichi, et al. "XML schema mappings: Data exchange and metadata management." Journal of the ACM (JACM) 61.2 (2014): 12.