

A Meta-Model for Development Process of IoT Application by Using UML

Eun-Sook Cho*, Chee-Yang Song**

Abstract

An Internet of Things(IoT) technology which provides intelligent services by combining context-awareness based intelligences, inter-communication is made of between things and things or between things and person through the network connected with intelligent things is spreading rapidly. Especially as this technology is converged into smart device, mobile, cloud, big data technologies, it is applied into various domains. Therefore, this is different from existing Web or Mobile Application. New types of IoT applications are emerging by adapting IoT into Web or mobile. Because IoT application is not only focused on software but also considering hardware or things aspect, there are limitations existing development process. Existing development processes don't consider analysis and design techniques considering both hardware and things. We propose not only a meta-model for development process which can support IoT application's development but also meta-models for main activities in this paper. Especially we define modeling elements by using UML's extension mechanisms, provide development process, and suggest design techniques how to apply those elements into IoT application's modeling phase. Because there are many types of IoT application's type, we propose an Android and Arduino-based on IoT application as a case study. We expect that proposed technique can be applied into many of various IoT application development and design with a form of flexible and extensible as well as main functionalities or elements are more concretely described. As a result, it brings IoT application's flexibility and the effect of quality improvement.

▶Keyword: IoT Technology, Development Process, Meta-Model, UML, Design Technique

I. Introduction

사물인터넷(Internet of Things, 약어로 IoT)은 각종 사물에 센서와 통신 기능을 내장하여 인터넷에 연결하는 기술 즉, 무선 통신을 통해 각종 사물을 연결하는 기술을 의미한다. 인터넷으로 연결된 사물들이 데이터를 주고받아 스스로 분석하고 학습한 정보를 사용자에게 제공하거나 사용자가 이를 원격 조정할 수 있는 인공지능 기술이다[1]. 이 기술은 스마트 디바이스, 모바일, 클라우드, 빅 데이터 등과 같은 기술들과 서로 융합하여 개방과 공유를 지향하는 초연결 사회의 핵심 기술로 급부상하고 있으며, 이는 신산업 창출, 생산성·효율성 제고, 사회 현안 해결 및 삶의 질 향상 등에 크게 기여할 것으로 전망되고 있다

[2]. 2013 가트너(Gartner) 보고서[3]에 따르면 2020년에 인터넷에 연결되는 사물의 수는 약 260억 개까지 증가할 것으로 전망되며(PC, 태블릿, 스마트폰 제외), 약 3,000 억 달러의 시장(서비스, 제품) 창출과 1.9조 달러의 경제적 파급 효과가 기대되며, 특히 제조와 헬스케어 분야의 파급 효과가 가장 클 것으로 전망하고 있다. 이미 이러한 전망이 우리 실생활의 여러 다양한 분야들에 적용되어 운영되고 있는 실정이다. 예를 들면, 스마트 디바이스에서 원격으로 집에 있는 여러 가전 기기를 제어 하는 스마트 홈을 비롯하여, 농작물에 적용되는 스마트 팜, 제조 분야의 스마트 공장, 스마트 그리드, 헬스케어 등 많은

• First Author: Eun-Sook Cho, Corresponding Author: Chee-Yang Song
*Eun-Sook Cho (escho@seoil.ac.kr), Dept. of Software Engineering, Seoil University
**Chee-Yang Song (cysong@knu.ac.kr), Dept of Software, Kyungbuk University
• Received: 2018. 12. 28, Revised: 2019. 01. 23, Accepted: 2019. 01. 25.
• The present research has been conducted by the Research Grant of Seoil University in 2018.

IoT 응용 서비스(애플리케이션)들이 존재한다. 이런 다양한 IoT 응용 서비스들을 개발하기 위해서는 체계적인 개발 프로세스의 정의와 각 단계별 필요한 분석 및 설계 기법들이 필요하다. 그런데 현재 IoT 애플리케이션에 특화 된 개발 프로세스나 분석 및 설계 기법들이 미비하게 제시된 수준이다. 따라서 본 논문에서 이러한 한계점들을 극복하고자 여러 다양한 형태의 IoT 애플리케이션 개발에 적용될 수 있는 프로세스와 UML 기반의 분석, 설계 기법들을 메타 모델 형태로 제시하고자 한다. 개발 프로세스나 각 단계별 기법들을 메타모델로 제시하는 이유는 IoT 애플리케이션 유형이 다양하기 때문에 여러 다양한 유형별로 메타모델을 기반으로 확장하여 적용할 수 있기 때문이다. 또한 현존 UML 모델링 요소들은 IoT 애플리케이션을 설계하는데 필요한 요소들이 부족하기 때문에 확장 메커니즘을 적용하기 위함이다.

본 논문의 구성으로 2장에서는 관련연구로 안드로이드 모바일 플랫폼과 아두이노 IoT 플랫폼, 그리고 MOF에 대해 설명하며, 3장에서는 본 논문에서 제안하는 IoT 애플리케이션 개발 프로세스 프레임워크와 메타모델들을 정의하여 제시한다. 4장에서는 3장에서 제안한 메타모델을 적용한 실제 모델 사례에 적용한 결과를 제시한다. 마지막으로 5장에서 결론 및 향후 연구 과제를 제시한다.

II. Preliminaries

1. Related works

1.1 Android Platform

IoT 응용 서비스와 연동하는 클라이언트 단말의 형태는 대부분 모바일 스마트 단말이다. 모바일 플랫폼의 종류에는 애플사에서 만든 IOS, 구글에서 만든 안드로이드, 마이크로소프트사에서 만든 윈도우즈 모바일, 노키아에서 만든 심비안(Symbian), 기타 등등 여러 종류가 있다[4,5]. 본 논문에서는 오픈 소스 기반의 오픈 플랫폼인 안드로이드 모바일 플랫폼을 적용한다.

안드로이드는 구글에서 만든 모바일 플랫폼으로서 다음과 같은 특징들을 지니고 있다[4,5,6].

첫째, 오픈 플랫폼이다. 안드로이드는 개방형 리눅스 커널을 기반으로 하고 있으며, 메모리와 하드웨어 리소스를 모바일 환경에 최적화시킨 맞춤형 버추얼(가상) 기기를 사용하고 있다. 따라서 모바일 앱들을 다양한 하드웨어에 이식하거나 새로운 기능을 부여하기가 용이하다.

둘째, 애플리케이션들의 결과물이 동일하다. 안드로이드는 안드로이드 플랫폼이 기본적으로 제공하는 내장 애플리케이션들과 SDK를 사용하기 때문에 개발자가 작성한 애플리케이션들의 결과물이 동일하다.

셋째, 애플리케이션 상호 호환성이 높다. 안드로이드는 새롭

고 혁신적인 애플리케이션에 대한 경계를 없애고 있다. 예를 들면, 개발자가 웹으로부터 정보를 가져와 사용자의 연락처들, 캘린더, 또는 지리적 위치 정보 같은 개인의 모바일 폰에 있는 데이터와 병합할 수 있다.

넷째, 애플리케이션 개발이 용이하다. 안드로이드는 풍부한 애플리케이션 설계를 위한 다양하고 유용한 라이브러리와 툴을 프레임워크 형태로 제공하기 때문에 애플리케이션 개발에 필요한 컴포넌트들이 프레임워크에서 많이 제공한다. 이는 높은 개발 생산성 효과를 준다.

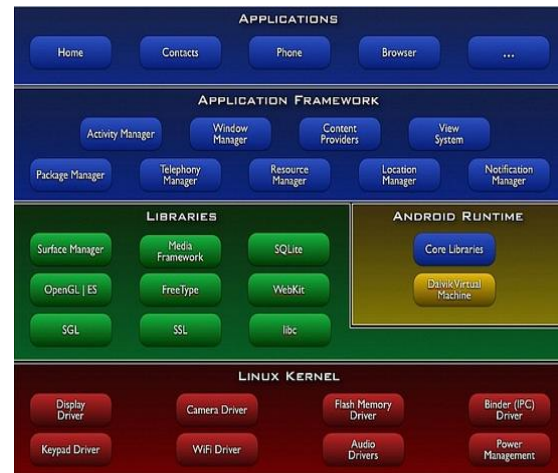


Fig. 1. Android Architecture

1.2 Arduin Platform

IoT 디바이스 플랫폼으로는 최근 각광을 받고 있는 ‘개방형 디바이스 플랫폼’인 아두이노(Arduino)와 라즈베리 파이(Raspberry Pi)가 주로 사용되고 있다[7,8]. 이는 하드웨어와 소프트웨어가 모두 오픈소스로 공개되어 있는데다 상대적으로 배우고 활용하기가 쉽기 때문에 기본적인 하드웨어 지식을 익힌 뒤 윈도우, 맥OS X, 리눅스 등 각종 운영체제에서 작동되는 개발 도구를 이용하면 누구든지 쉽게 개발할 수 있기 때문이다.

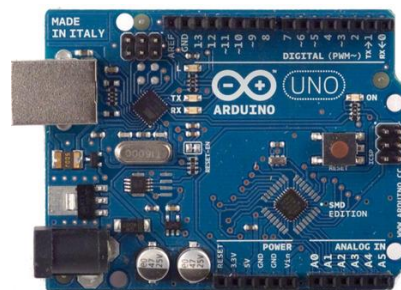


Fig. 2. Arduino UNO

1.3 MOF(Meta Object Facility)

메타모델이란 어떠한 개념들을 구성하는 주요 요소들과 그들 간의 관계를 보여주는 개념 맵이다.

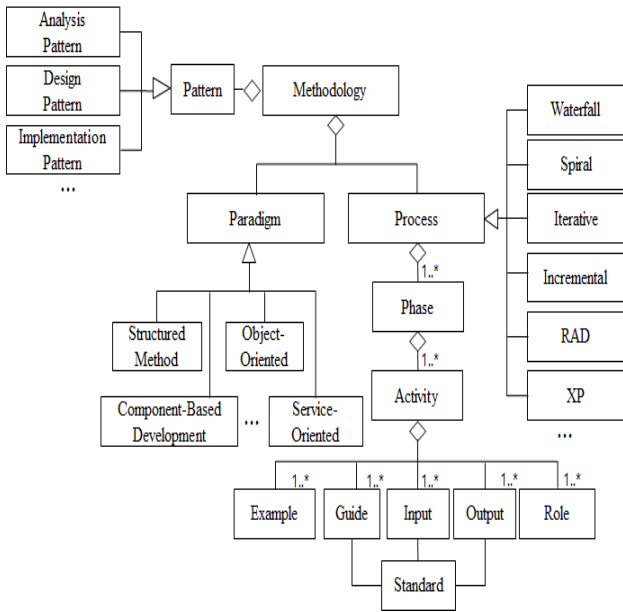


Fig. 3. Meta-model of Development Process

예를 들면 객체지향 모델링 언어인 UML(Unified Modeling Language)의 요소들에 대한 의미를 모델로 정의한 MOF(Meta Object Facility)가 메타 모델에 해당한다[9,10]. 또한 MDA에서는 이 MOF를 CWM(Common Warehouse Metamodel)이나 UML의 메타모델에 대한 공통 모델로 제공하고 있다[11]. 메타모델은 특정 개발 플랫폼에 종속되지 않은 형태로 제공되기 때문에 메타 모델에 기반을 두고 모델을 개발할 경우 플랫폼에 따른 다양한 형태의 모델들을 확장하여 개발할 수 있을 뿐만 아니라 모델의 재사용성이 매우 향상되게 된다. 따라서 본 논문에서도 이 MOF를 IoT 애플리케이션 개발 프로세스와 분석 및 설계 모델을 설계하는데 사용하고자 한다.

III. The Proposed Scheme

1. Design of Meta-Model for IoT Application's Development Process

이 장에서는 안드로이드와 아두이노 기반의 IoT 애플리케이션 개발을 위한 개발 프로세스를 메타모델을 이용해서 설계한다. 이러한 개발 프로세스 메타모델을 적용하여 개발하게 되면 개발 생산성을 높일 수 있을 뿐만 아니라 개발된 IoT 응용 서비스의 독립성, 재사용성 및 품질이 향상되게 된다.

1.1 Meta-model of Methodology's Framework

본 논문에서는 IoT 애플리케이션 개발 프로세스 메타모델을 [12]에서 제시한 MOF를 반영하여 클래스 모델로 표현하여 정의한다. 소프트웨어 공학에서 하나의 개발 방법론은 패러다임과 프로세스로 구성된다고 정의한다. 여기서 패러다임이란 예를 들면 절차지향, 객체지향, 컴포넌트 기반, 서비스 지향 등과 같은 것을 말한다. 그리고 프로세스란 폭포수 모델(Waterfall Model), 나선형 모델(Spiral Model), 반복형 모델(Iterative Model), 점진형 모델(Incremental Model), RAD형 (Rapid Application Development), XP, Agile 등과 같은 것을 말한다.

그리고 하나의 개발 프로세스는 여러 단계(Phase)들로 구성되며, 각각의 단계는 여러 활동(Activity)들로 구성되고, 또 각 활동은 여러 지침(Instruction)들로 구성된다. 이러한 요소들을 반영하여 MOF를 기반으로 설계한 메타모델이 그림 3에 제시되어 있다.

이러한 메타모델을 기반으로 본 논문에서는 IoT 애플리케이션 개발 프로세스를 크게 3개의 단계인 IoT 응용 서비스 분석, IoT 응용 서비스 설계, IoT 응용 서비스 구현 단계로 정의하였다.

1.2 Meta-Model for IoT Application's Development Process

본 논문에서는 그림 2의 소프트웨어 개발 프로세스 메타모델을 기반으로 IoT 애플리케이션 개발 프로세스 모델을 생성하여 그림 4와 같이 정의하였다.

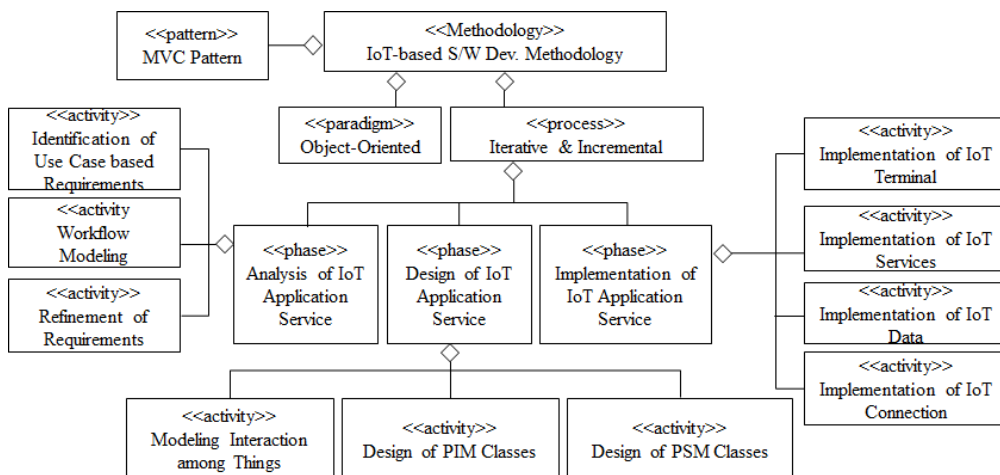


Fig. 4. Meta-model of IoT Application's Development Process

생성된 개발 프로세스는 크게 3개의 단계인 IoT 응용 서비스 분석, IoT 응용 서비스 설계, IoT 응용 서비스 구현 단계로 나뉜다. 각 단계별로 활동 또는 업무들로 구성되어 있다

예를 들면 IoT 응용 서비스 분석 단계는 3개의 업무인 Use Case 기반 요구사항 식별, 워크플로우 모델링, 요구사항 정제 등으로 구성되어 있다. 또한 각 활동 별로 필요한 모델링 요소들이 메타모델로 정의된다.

1.3 Meta-model of Activities

앞에서 정의된 모바일 앱 개발 프로세스 메타모델에서 각 활동에 필요한 모델링 요소들에 대해서도 메타모델로 정의할 수 있다. 이 절에서는 이러한 각 활동별 메타모델을 정의한다.

1.3.1 Identification of Use Case based Requirements

유스 케이스 기반의 요구사항 식별 활동에서는 IoT 애플리케이션 개발에 필요한 요구사항들을 유스 케이스 단위로 식별한다. 요구사항들 가운데 기능 요구사항을 보다 쉽게 식별하기 위해 UML의 유스 케이스 다이어그램을 이용한다.

이에 대한 메타모델이 그림 5에 제시된다.

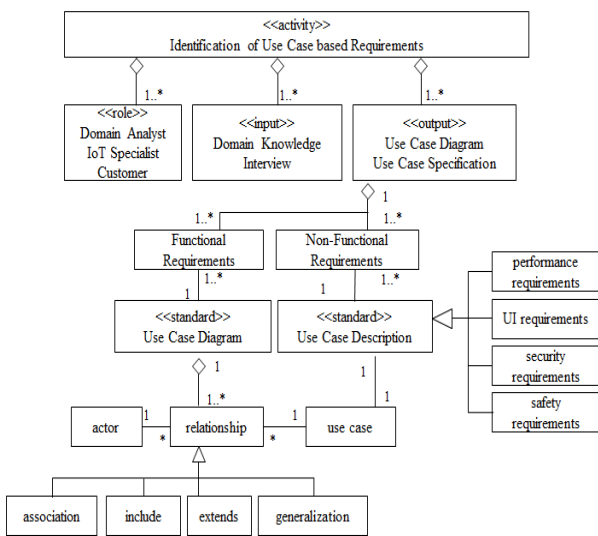


Fig. 5. Meta-model of Identifying Requirements based on Use Case

1.3.2 Workflow Modeling

워크플로우 모델링은 개발한 IoT 애플리케이션의 업무 흐름을 명세한다. 앞서 추출한 유스케이스들은 개발한 애플리케이션에 필요한 기능들만 표현했지, 이들 간의 순서는 표현이 되지 않았다. 따라서 여기서 추출한 유스케이스들을 토대로 이들의 실행 순서를 워크플로우로 모델링한다.

워크플로우는 UML의 활동도(Activity Diagram)을 이용하여 모델링하는데, 그림 6에 제시된다.

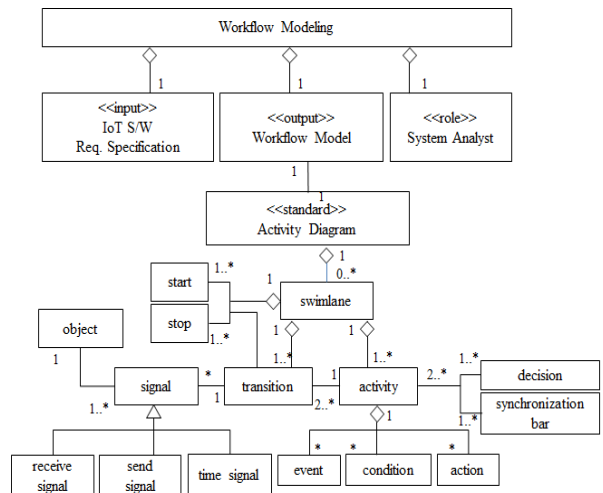


Fig. 6. Meta-model of Workflow Modeling

1.3.3 Modeling Interactions among Things

사물 간 상호 작용도는 UML의 순차도(Sequence Diagram)를 응용한 다이어그램으로서, 시스템 내의 객체들뿐만 아니라 IoT 사물과의 상호작용을 표현한다.

앞서 작성된 유스 케이스 다이어그램을 입력물로 받아서 액터와 객체, 객체와 객체, 객체와 사물 간의 상호작용을 표현하는 모델이다. 이 사물 간 상호작용도의 객체 유형을 보면 기존 객체 이외에 IoT 사물 객체가 확장되어 반영됨을 알 수 있다. 사물 간 상호 작용 모델링에 대한 메타 모델은 그림 7과 같다.

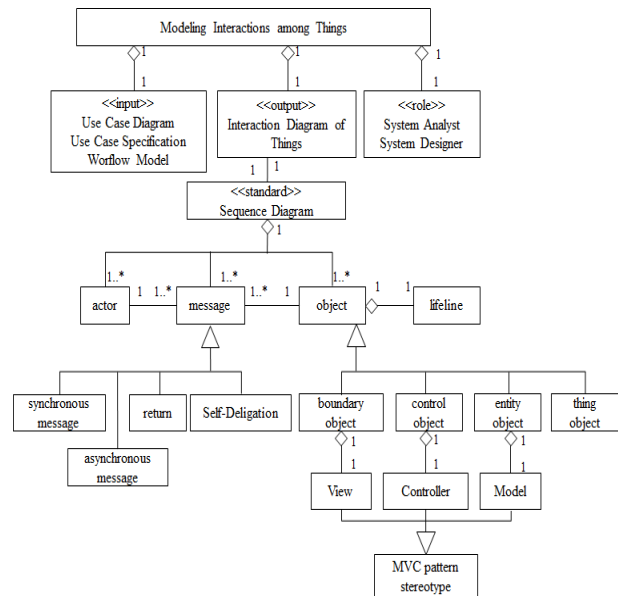


Fig. 7. Meta-model of Modeling Interactions among Things

1.3.4 Design of PIM(Platform Independent Model) Class

앞서 정의한 사물 간 상호 작용도를 입력물로 받아서 전체 IoT 애플리케이션 필요한 클래스들을 추출하고 클래스들 간의 관계를 맺어준다. 그리고 클래스의 유형은 MVC를 기반으로 하기 때

문에 크게 3가지 유형으로 분류하고, 클래스에는 UML(Unified Modeling Language)의 스테레오타입을 이용하여 명시한다. 이에 대한 메타모델이 그림 8에 제시되어 있다.

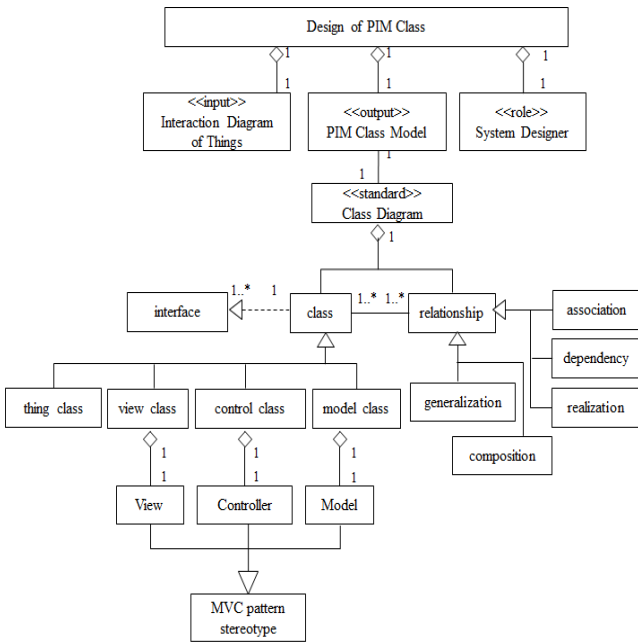


Fig. 8. Meta-model of Designing PIM Class

1.3.5 Design of PSM(Platform Specific Model) Class

PIM 클래스 모델이 설계되면 이를 입력물로 받아서 안드로이드 플랫폼과 아두이노 IoT 플랫폼의 스테레오타입을 반영한 PSM 클래스 다이어그램을 작성한다.

PIM 클래스 모델은 플랫폼에 독립적인 모델이 되는 반면에 그림9처럼 PSM 클래스 모델은 구현할 플랫폼이 반영된 클래스 모델이기 때문에 안드로이드나 아두이노에 특화된 클래스들이 추가되어 확장된 모델이 된다. 이처럼 PIM 모델과 PSM 모델을 구분하여 설계하는 이유는 모델의 재사용성과 독립성, 그리고 확장성 등을 높이기 위함이다. 이뿐만 아니라 자동화 도구로 구현함으로써 PIM 모델을 설계하기만 하면 PSM 모델은 적용할 플랫폼을 선택하면 자동으로 모바일 앱 코드가 생성될 수 있도록 구축할 수 있다. 이는 결국 특정 플랫폼에 최적화된 IoT 애플리케이션 구현을 보다 매끄럽게 연결해 준다.

IV. Experiment

이 장에서는 본 논문에서 제시하는 메타모델들을 기반으로 실제 IoT 애플리케이션 개발 프로세스에 적용하여 생성된 모델들에 대한 사례 연구를 제시 하고자 한다.

1. Case Study

본 논문에서는 안드로이드 앱을 단말로 해서 원격으로 온도를 제어하는 시스템을 사례 연구로 채택하였다[11].

1.1 Identification of Use Case based Requirements

그림 4의 메타모델을 적용하여 메모 앱의 기능 요구사항들을 식별하여 그림 10과 같이 유스 케이스 다이어그램으로 표현하였다.

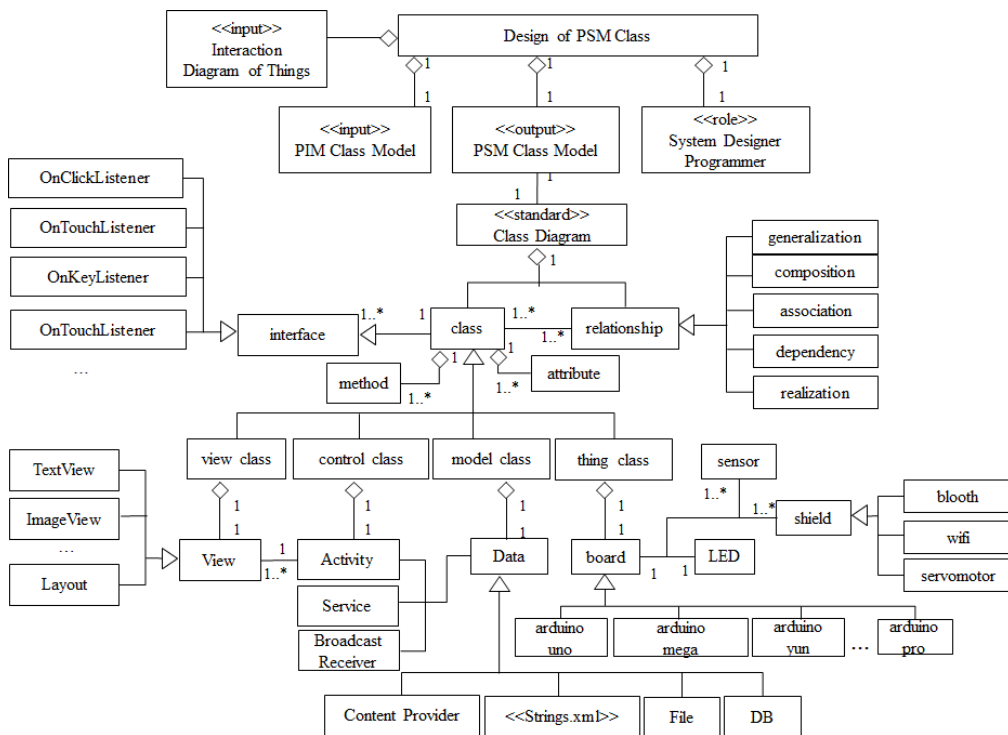


Fig. 9. Meta-Model of Designing PSM Class

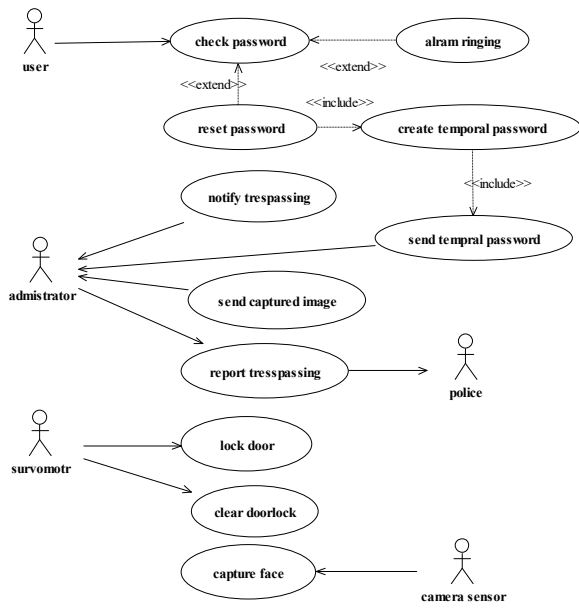


Fig. 10. Use Case Model

1.2 Workflow Modeling

식별된 기능 요구사항들에 대한 각각의 유스 케이스에 대해 각각의 유스 케이스 내의 사건 흐름을 모델링하기 위해 워크 플로우 메타 모델을 적용하여 그림 11과 같이 작성하여 애플리케이션의 업무 흐름을 명세하였다.

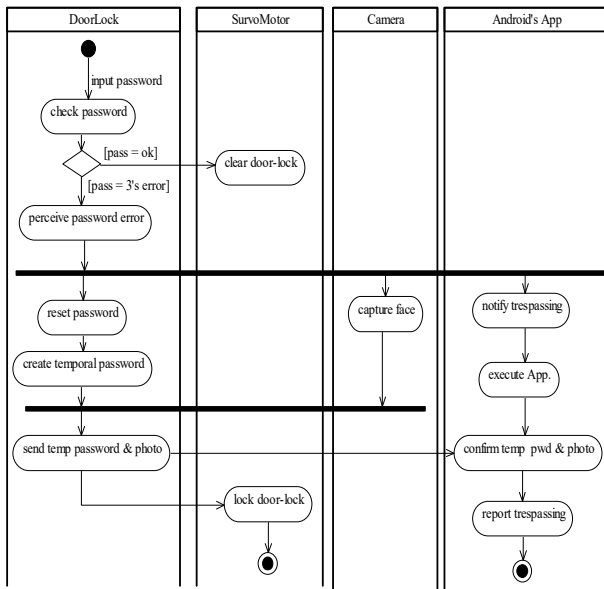


Fig. 11. Workflow Model

1.3 Modeling Interactions among Things

작성된 시스템 순서도를 입력물로 받아서 앱 내에 참여하는 객체들을 MVC 패턴을 적용하여 3가지 형태의 클래스(또는 객체)들로 분류하여 UML의 상호 작용도를 이용하여 객체 간의 상호작용을 설계하였다. 그림 12는 그림 7의 메타 모델을 적용하여 생성된 결과이다.

1.4 Design of PIM Class

그림 11의 사물 상호 작용도를 입력물로 받아서 클래스들을 정의하고, 클래스에 필요한 속성, 메소드를 정의하고, 클래스들 간의 관계를 그림 8의 메타모델을 적용하여 설계하였다.

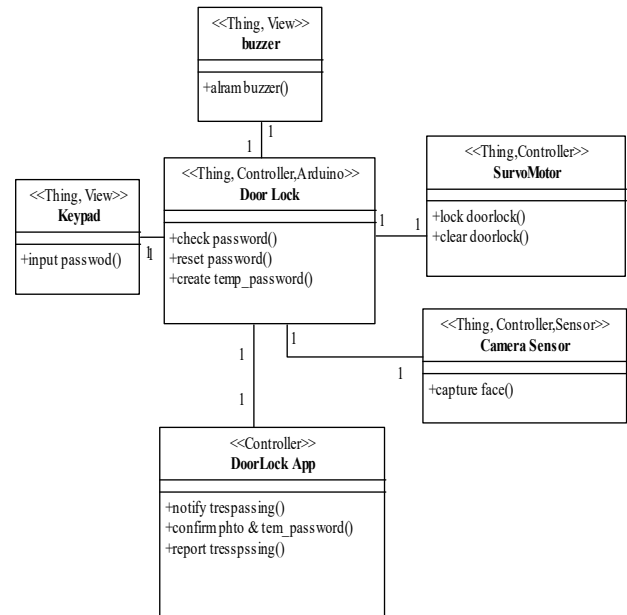


Fig. 13. PIM Class Model

1.5 Design of PSM Class

그림 12의 상호 작용도와 그림 13의 PIM 클래스 다이어그램을 입력물로 받아서 플랫폼과 IoT 특징들을 반영하여 클래스 다이어그램을 정제하면 그림 14와 같다.

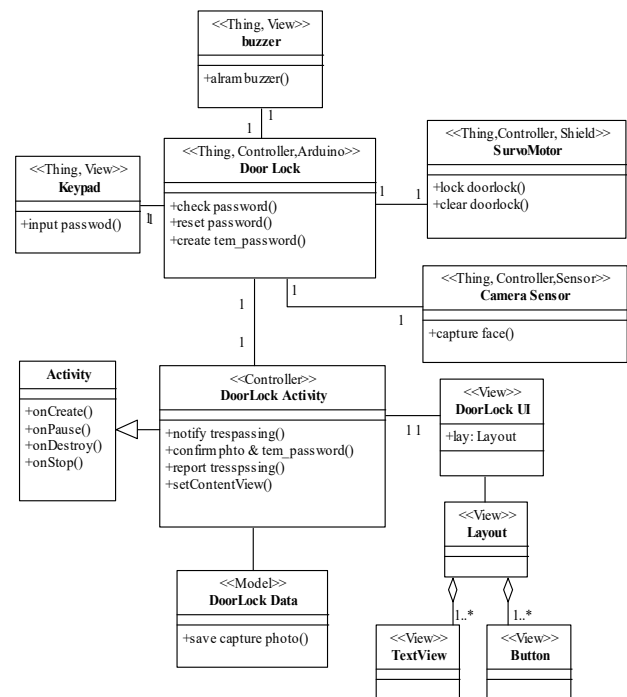


Fig. 14. PSM Class Model

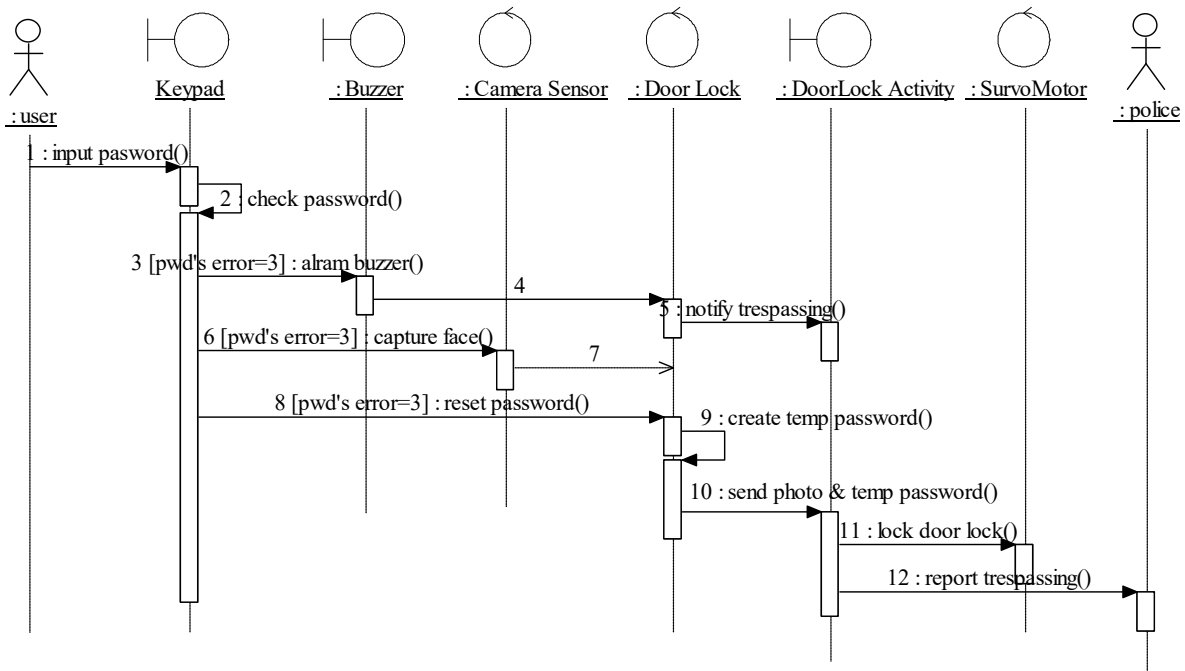


Fig. 12. Interaction Diagram among Things

사물 간 상호 작용도에 표현된 클래스들 가운데 안드로이드 앱 클래스는 다시 화면은 뷰 클래스로 매핑되고, 로직 처리 부분은 Activity 클래스로 매핑되고, 도어락으로부터 전송 받은 Data는 클래스로 매핑된다. 이를 바탕으로 그림 9의 PSM 메타 모델을 적용하여 정제된 PSM 클래스 모델은 그림 14와 같다.

구현단계에 있어서 보다 자동 코드 생성으로의 구현이 가능하게 된다. 이는 결국 IoT 애플리케이션 자동 생성 메커니즘을 위한 기반이 되는 것이다. 따라서 향후 연구 과제로는 제시한 메타모델을 기반으로 자동화 도구를 구축하는 것이다.

IV. Conclusions

IoT 기술이 여러 다양한 분야에 융합되어 개발되고 있는 반면에 체계적인 IoT 애플리케이션 개발 프로세스나 모델링 기법은 거의 제시되고 있지 못한 상황이다. 이런 상황에서 개발된 IoT 애플리케이션들은 모듈의 독립성, 재사용성, 확장성 등과 같은 품질을 측정했을 경우, 그 품질에 대한 보장이 어렵다. 이는 결국 애플리케이션의 신뢰도가 저하되는 결과를 초래한다. 특히 IoT 애플리케이션은 사물과의 연동이 이루어지기 때문에 품질에 대한 요구사항이 더 높다고 볼 수 있다. 본 논문에서는 이러한 문제점을 해결하고자 IoT 애플리케이션 개발에 있어서 일관성 있으면서 체계적이고 보다 품질 향상을 높이기 위한 개발 프로세스를 메타모델로 설계하였다.

본 논문에서 IoT 애플리케이션 개발 프로세스를 메타모델로 제시한 이유는 여러 다양한 유형의 IoT 애플리케이션 개발에 있어서 그 목적에 맞게 커스터마이제이션(Customization)할 수 있도록 하기 위함이다. 특히 본 논문에서는 안드로이드 기반의 모바일 앱과 아두이노 IoT 플랫폼을 기반으로 한 IoT 애플리케이션 개발에 맞춰서 MVC 패턴과 UML을 접목시킨 메타모델을 제시하였다. 이는 모델링에서부터 MVC 패턴을 적용함으로써

REFERENCES

- [1] IDC's Worldwide Internet of Things (IoT) Taxonomy, October, 2013.
- [2] Y.K. Chen, "Challenges and Opportunities of Internet of Things", In Proceedings of the 17th Asia and South Pacific Design Automation Conference(ASP-DAC 2012), pp.383-388, Jan., 2012.
- [3] The Internet of Things, Worldwide, Gartner, Inc. Nov. 2013.
- [4] Google Android [Online]. <http://www.android.com>
- [5] Salmre, I., Writing Mobile Code: Essential Software Engineering for Building Mobile Applications, Addison-Wesley Professional, 2005.
- [6] S. Kim, Android Programming Complete Guide, Hanbit-Media, 2011.
- [7] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey", Computer Networks, Vol. 54, pp.2787-2805, 2010.
- [8] A. Gluhak, S. Krco, M. Nati, D. Pfistere, N. Mitton, and T. Razafindralambo, "A Survey on Facilities for Experimental Internet of Things Research," IEEE

- Communication Magazines, Vol.49, No.11, pp.58-67, Nov., 2011.
- [9] Object Management Group. MOF Documents[Online], <http://www.omg.org/spec/MOF,2018>.
- [10] R. Pressman, B. Maxim, Software Engineering : A Practitioner's Approach, 8th Edition, McGraw-Hill Science/Engineering/Math, pp.187~198, Jan., 2014.
- [11] H. J. La, S. D. Kim, "Unconventional Issues and Solutions in Developing IoT Applications," KIPS Tr. Comp. and Comm. Sys., Vol.3, No.10, pp.337-350, Mar., 2014.
- [12] Eun Sook Cho, "Design of Meta-model for the Development Process of a Mobile Application", Vol.15, No.8, pp.5248-5255. 2013.

Authors



Eun-Sook Cho received the B.S. degree in Computer Science from DongEui University, Korea in 1993. He received the M.S and Ph.D degree in Computer Science from SoongSil University, Korea, in 1996 and 2000, respectively. Dr. Cho joined the

faculty of the Department of Software Engineering at Seoil University, Seoul, Korea, in 2005. He is currently a Professor in the Department of Software Engineering, Seoil University. He is interested in component-based Development, cloud computing, and IoT Applications.



Chee-Yang Song received the B.S. degree in Computer Science from Hannam University, Korea in 1985. He received the M.S degree in Computer Science from Chung-Ang University, Korea, in 1987, and received the Ph.D degree in Dept of

Computer Science from Korea University, Korea, in 2003. Dr. Song joined the faculty of the Department of software at Kyungbuk University, Sangju, Korea, in 2005. He is currently a Professor in the Department of Software, Kyungbuk University. He is interested in component-based Development, cloud computing, and IoT Applications.