

모바일 앱 개발 프로세스 메타모델 설계

조은숙*

¹서일대학교 컴퓨터 소프트웨어과

Design of a Metamodel for the Development Process of a Mobile Application

Eun-Sook Cho*

¹Dept. of Computer Software, Seoil University

요약 스마트 폰 시장의 급성장으로 인해 모바일 앱들이 우후죽순처럼 개발되고 있는 실정이다. 이렇게 급격한 속도로 모바일 앱들이 개발되다 보니 체계적인 개발 프로세스의 적용이나 재사용성 및 확장성 등과 같은 품질 요소들고려하지 않은 채 개발하는 경우가 허다하다. 결국 이렇게 개발된 모바일 앱들은 내부적으로 여러 결함 요소들을 갖게 되기 때문에 사용자들로 하여금 만족도나 신뢰도 부분에서 매우 낮은 평가를 받게 되고 결국 사용자들로 하여금 앱에 대한 업데이트나 재구매를 포기하게끔 한다. 따라서 본 논문에서 보다 사용자들의 만족도와 신뢰도를 높이고 모바일 앱의 지속적인 사용 효과를 높이기 위해 모바일 앱 개발 프로세스를 위한 메타모델을 설계하여 정의하고자한다. 여러 다양한 모바일 앱 개발에 있어서 메타 모델을 적용한 개발 프로세스를 적용하게 되면 개발 생산성을 높일 수 있을 뿐만 아니라 모바일 앱의 품질 향상의 효과도 가져올 수 있다.

Abstract Because of the rapid growth of smart phone market, the number of mobile applications has increased exponentially. As mobile applications are developing at a rapid rate, many of them are being developed without considering a systematic development process or quality factors, such as reusability, extensibility, etc.. As a result, they have many faults internally, they are being underrated in satisfaction or reliability from customers, and finally customers do not update or repeat the purchase. Therefore, a meta-model of development process for mobile applications was designed to improve the continuous usability of mobile applications as well as the satisfaction and reliability from users. Applying development process based on a meta-model to mobile application development is expected to improve development productivity and quality of mobile applications.

Key Words : Mobile App., Development Process, Meta-Model, Reusability, Productivity

1. 서론

전 세계적으로 스마트폰 시장의 규모는 나날이 급격하게 거대해지고 있다. 미국의 시장조사업체 스트래티지 애널리틱스(SA)의 보고에 따르면 올해 스마트폰이 세계 시장에서 지난해보다 21.2% 늘어 12억10만대 팔릴 것이라고 전망하고 있다[1]. 이와 같은 세계 스마트폰 시장 규모는 지난 2007년 1억1천970만대에서 7년만에 10배 가량

늘어난 결과에 이른 것이다. 이처럼 스마트폰 시장의 규모가 늘어나면서 이와 동반하여 스마트폰에서 작동되는 모바일 앱들에 대한 개발도 우후죽순처럼 늘어나고 있는 상황이다. 특히 여러 모바일 플랫폼에 포팅할 수 있는 앱들을 개발하려고 하다 보니 모바일 앱 개발 시에 체계적인 개발 프로세스의 적용이나 품질 요소에 대한 고려가 잘 반영되지 않은 채 개발되는 경우들이 많아지고 있다. 그런데 이러한 앱들이 실제 사용하는데 있어서 기능의

본 논문은 2014년 서일대학교 교내연구과제로 수행되었음.

*Corresponding Author : Eun-Sook Cho(Seoil Univ.)

Tel: +82-10-8870-4595 email: escho@seoil.ac.kr

Received May 27, 2014

Revised June 11, 2014

Accepted August 7, 2014

오작동, 사용자 인터페이스의 불편함, 예기치 못한 결함 발생 등의 여러 가지 문제점들을 야기 시키고 있다. 이는 결국 사용자들로 하여금 구매한 모바일 앱에 대한 만족도나 신뢰도를 저하시킬 뿐만 아니라 재구매 혹은 갱신을 포기하게끔 만들고 있다.

이러한 문제점들이 개선되지 않은 채 계속적으로 개발되다 보면 결국 모바일 앱 시장은 점차 사장될 수도 있다. 본 논문에서 이러한 문제점을 해결하기 위해 모바일 앱 개발을 하는 데 있어서 보다 개발 생산성을 높이고 모바일 앱의 품질을 향상시키기 위해 메타 모델 기반의 개발 프로세스 프레임워크를 제시하고자 한다. 이러한 기법을 모바일 앱 개발시에 적용함으로써 체계적인 모바일 앱 개발이 이루어질 수 있을 뿐만 아니라 재사용성, 독립성, 확장성이 있는 고품질의 모바일 앱 개발이 이루어지리라 기대한다.

본 논문의 구성으로 2장에서는 관련연구로 모바일 플랫폼과 메타 모델 언어인 MOF에 대해 설명하며, 3장에서는 본 논문에서 제안하는 모바일 앱 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.

2. 관련 연구

모바일 플랫폼이란 모바일 기기와 사용자를 연결해주는 플랫폼을 말한다. 간단히 말하면 모바일 운영체제라 할 수 있다. 모바일 플랫폼의 종류에는 애플사에서 만든 IOS, 구글에서 만든 안드로이드, 마이크로소프트사에서 만든 윈도우즈 모바일, 노키아에서 만든 심비안(Symbian), 기타 등등 여러 종류가 있다[2,3]. 현재 이러한 종류들 가운데 전 세계적으로 가장 많이 사용하는 것은 애플의 IOS와 구글의 안드로이드 플랫폼이다. 본 논문에서는 오픈 소스 기반의 오픈 플랫폼인 안드로이드 기반의 모바일 앱 설계 및 개발 기법을 제시한다. 따라서 이 장에서는 안드로이드 플랫폼의 특징과 MDA의 메타 모델 정의언어인 MOF에 대해 설명하고자 한다.

2.1 안드로이드 플랫폼의 특징

안드로이드는 구글에서 만든 모바일 플랫폼으로서 다

음과 같은 특징들을 지니고 있다[3,4,5].

■ 오픈 소스 플랫폼

안드로이드는 개방형 리눅스 커널을 기반으로 하고 있다. 나아가, 그것은 메모리와 하드웨어 리소스를 모바일 환경에 최적화시킨 맞춤형 버추얼(가상) 기기를 사용하고 있다. 따라서 모바일 앱들을 다양한 하드웨어에 이식하거나 새로운 기능을 부여하기 용이하다.



[Fig. 1] Android Architecture

■ 애플리케이션들의 동질성

안드로이드는 안드로이드 플랫폼이 기본으로 제공하는 내장 애플리케이션들과 SDK를 사용하여 개발자가 작성한 애플리케이션들의 결과물이 동일하다. 예를 들면 전화기의 경우, 다양한 종류의 애플리케이션과 서비스를 사용자에게 제공하는 전화기의 성능에 동일한 접속이 가능하도록 설계되었다. 안드로이드 플랫폼 기반의 기기로서, 사용자들은 자신의 기호에 따라 전화기를 맞춤화 할 수 있다. 전화기 대기화면을 바꿀 수도 있고, 스타일을 바꿀 수도 있으며, 애플리케이션의 종류도 바꿀 수 있다. 심지어는 전화기를 통해 가장 선호하는 사진 보기 애플리케이션을 사용해 모든 사진을 보게 하도록 설정할 수 있다. 이는 모바일 기기의 모든 자원을 최대한 활용할 수 있는 강력한 애플리케이션을 만들어 낼 수 있다.

■ 애플리케이션 간의 상호 호환성

안드로이드는 새롭고 혁신적인 애플리케이션에 대한 경계를 없앱니다. 예를 들자면, 좀 더 사용자 중심적인 사용감을 제공하기 위해서 개발자는 웹으로부터 정보를 가져와 사용자의 연락처들, 캘린더, 또는 지리적 위치 정보

같은 개개인의 모바일 폰에 있는 데이터와 병합할 수 있다. 안드로이드라면, 개발자는 사용자들에게 친구들의 위치를 보여주고 알려주어 근처에 있을 때 연락할 수 있는 기회를 제공하는 것이 가능한 애플리케이션도 설계할 수 있다.

■ 애플리케이션 개발의 용이성

안드로이드는 풍부한 애플리케이션 설계를 위한 다양하고 유용한 라이브러리와 툴을 제공한다. 일례로, 안드로이드는 개발자가 기기의 위치를 알게 하고, 기기 간에 통신할 수 있게 하여 뛰어난 피어-투-피어(Peer-to-Peer) 소셜 애플리케이션(Social Application)을 구현할 수 있다. 이뿐만 아니라 안드로이드는 개발자들로 하여금 높은 개발 생산성 효과를 주기 위해 통합 툴 세트를 포함하고 있다.

2.2 MOF(Meta Object Facility)

메타란 사전적 의미로 다른 개념으로부터의 추상화를 가리키며, 후자를 완성하거나 추가하는 데 쓰인다[6]. 따라서 메타모델이란 어떤 모델에 대한 의미를 추상화하여 표현한 모델로서, 예를 들면 객체지향 모델링 언어인 UML의 요소들에 대한 의미를 모델로 정의한 MOF(Meta Object Facility)가 메타 모델에 해당한다[7,8]. 본 논문에서도 이 MOF를 모바일 앱 개발 프로세스 프레임워크를 설계하는데 사용하고자 한다.

MOF는 객체 및 컴포넌트 기술의 핵심을 정형화한 모델로서 MDA에서 MOF는 CWM(Common Warehouse Metamodel)이나 UML(Unified Modeling Language)의 메타모델에 대한 공통 모델로서 제공된다[7,8,9,10]. MOF의 정형화된 내용은 MDA 모델들을 위한 표준 저장소의 역할을 수행한다.

3. 모바일 앱 개발 프로세스에 대한 메타모델 설계

이 장에서는 안드로이드 기반의 모바일 앱 개발을 위한 개발 프로세스를 메타모델을 이용해서 설계하고자 한다. 이러한 개발 프로세스 메타모델을 적용하여 개발하게 되면 개발 생산성을 높일 수 있을 뿐만 아니라 개발된 모바일 앱의 독립성, 재사용성, 그리고 확장성 또한 향상

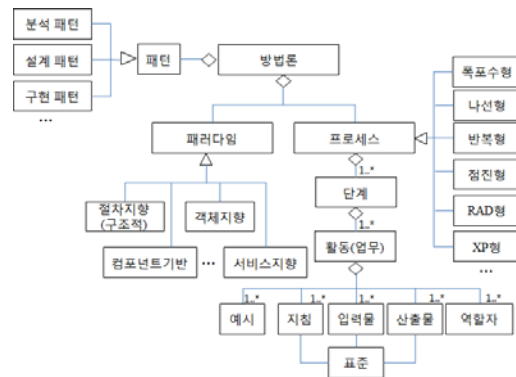
되게 된다.

3.1 개발 방법론 프레임워크 메타모델

본 논문에서는 모바일 앱 개발 프로세스 메타모델을 MOF를 반영하여 클래스 모델로 표현하여 정의한다. 소프트웨어 공학에서 하나의 개발 방법론(Methodology)은 개발 패러다임(Paradigm)과 개발 프로세스(Development Process)로 구성된다고 정의한다. 여기서 개발 패러다임이란 예를 들면 절차지향, 객체지향, 컴포넌트 기반, 서비스 지향 등과 같은 것을 말한다. 그리고 개발 프로세스란 폭포수 모델(Waterfall Model), 나선형 모델(Spiral Model), 반복형 모델(Iterative Model), 점진형 모델(Incremental Model), RAD형(Rapid Application Development) 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 Fig. 2에 제시되어 있다.

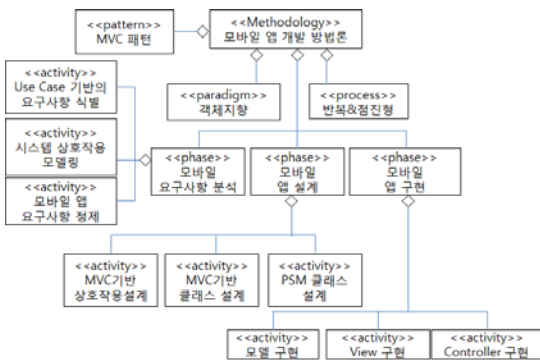


[Fig. 2] Metamodel of Development Process's Framework

이러한 메타모델을 기반으로 본 논문에서는 모바일 앱 개발 프로세스를 크게 3개의 단계인 모바일 앱 요구사항 분석, 모바일 앱 설계, 모바일 앱 구현 단계로 정의하였다.

3.1 모바일 앱 개발 프로세스 메타모델

본 논문에서는 Fig. 2의 소프트웨어 개발 프로세스 메타모델을 기반으로 모바일 앱 개발 프로세스 모델을 생성하여 Fig. 3과 같이 정의하였다.



[Fig. 3] Metamodel of Mobile App's Development Process

생성된 모바일 앱 개발 프로세스는 크게 3개의 단계인 모바일 앱 요구사항 분석, 모바일 앱 설계, 모바일 앱 구현 단계로 나뉘어 진다. 그리고 각 단계별로 활동 또는 업무들로 구성되어 있다. 예를 들면 모바일 앱 설계 단계는 3개의 업무인 MVC 기반 상호 작용 설계, MVC 기반 클래스 설계, 그리고 PSM 클래스 설계 업무들로 구성되어 있다. 또한 각 활동 별로 필요한 모델링 요소들이 메타모델로 정의된다.

3.2 활동 별 메타모델

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

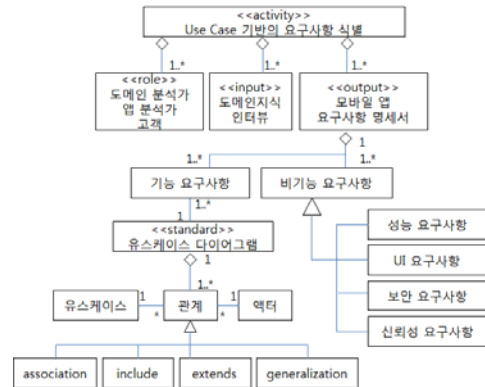
3.2.1 유스 케이스기반의 요구사항 식별

유스 케이스 기반의 요구사항 식별 활동에서는 모바일 앱 개발에 필요한 요구사항들을 식별한다. 요구사항들 가운데 기능 요구사항을 보다 쉽게 식별하기 위해 UML의 유스케이스 다이어그램을 이용한다. 이에 대한 메타모델이 Fig. 4에 제시된다.

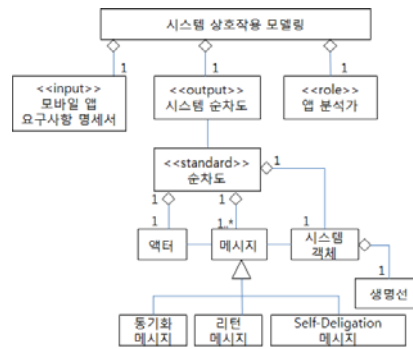
3.2.2 시스템 상호작용 모델링

시스템 순차도는 UML의 순차도(Sequence Diagram)을 응용한 다이어그램으로서, 시스템 내의 객체들 간의 상호작용을 모델링하기 전에 이전에 작성된 유스 케이스 다이어그램을 입력물로 받아서 액터와 시스템 간의 상호 작용 흐름을 표현하기 위한 모델이다. 이 시스템 순차도를 작성함으로써 이후 객체들 또는 컴포넌트들 간의 상

호 작용도를 작성하기가 훨씬 용이하다. 시스템 상호작용 모델링 활동에 대한 메타 모델은 Fig. 5와 같다.



[Fig. 4] Metamodel of Identifying Use Case-based Requirements



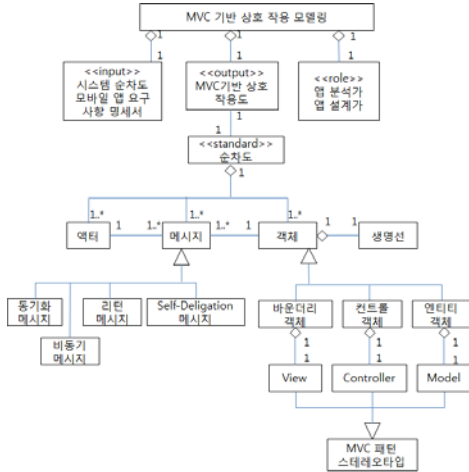
[Fig. 5] Metamodel of Modeling System Interaction

3.2.3 MVC 기반 상호 작용 설계

시스템 순차도에는 '앱(App.)' 하나가 하나의 객체로 매핑되어 표현된다. 여기서 이 앱이란 것은 개발될 모바일 앱 시스템 전체를 의미한다. 그러나 하나의 앱은 하나 이상의 뷰(또는 화면, UI)객체들과 제어 객체 및 데이터 객체들로 구성된다.

이러한 각각의 역할들을 담당하는 객체들이 서로 상호작용하며 기능들을 수행하게 된다. 따라서 총체적인 단위로 표현된 앱 객체를 보다 세분화 시켜서 표현해야 한다. 따라서 작성한 시스템 순차도의 시스템 객체가 수행한 메시지들을 그 메시지의 성격에 따라 MVC 패턴을 적용하여 뷰 객체(바운더리 객체), 컨트롤러 객체, 데이터 객체(모델 객체) 등으로 분할하여 메시지들을 해당 객체들로 할당한다. 이를 객체 또는 컴포넌트 간 상호 작용

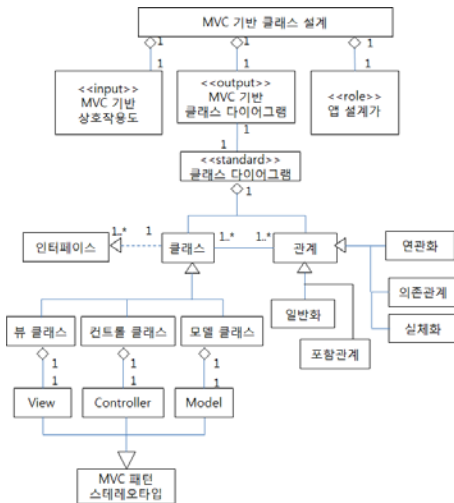
도(Interaction Diagram)라 한다. 이 상호 작용도에 대한 메타 모델이 Fig. 6에 제시되어 있다.



[Fig. 6] Metamodel of Modeling MVC-based Interaction Diagram

3.2.4 MVC 기반 클래스 설계

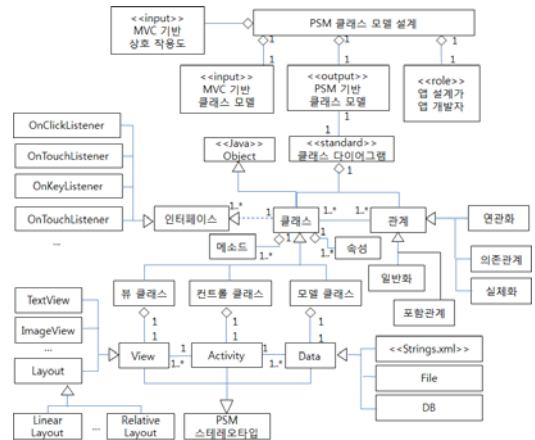
앞서 정의한 상호 작용도를 입력물로 받아서 전체 모바일 앱에 필요한 클래스들을 추출하고 클래스들 간의 관계를 맺어준다. 그리고 클래스의 유형은 MVC를 기반으로 하기 때문에 크게 3가지 유형으로 분류하고, 클래스에는 UML(Unified Modeling Language)의 스테레오타입을 이용하여 명시한다. 이에 대한 메타모델이 Fig. 7에 제시되어 있다.



[Fig. 7] Metamodel of Designing MVC-based Class

3.2.5 PSM(Platform Specific Model) 클래스 설계

MVC 기반으로 클래스 모델이 설계되면 이를 입력물로 받아서 안드로이드 플랫폼의 스테레오타입 패턴을 반영한 PSM 클래스 다이어그램을 작성한다. MVC 기반의 클래스 모델은 플랫폼에 독립적인 PIM(Platform Independent Model)이 되는 반면에 PSM 클래스 모델은 구현할 플랫폼이 반영된 클래스 모델이기 때문에 PSM 모델이 되는 것이다. 이처럼 PIM 모델과 PSM 모델을 구분하여 설계하는 이유는 모델의 재사용성과 독립성, 그리고 확장성 등을 높일 수 있다. 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 최적화 된 모바일 앱 구현을 보다 매끄럽게 연결해 준다.



[Fig. 8] Metamodel of Designing PSM Class

4. 평가

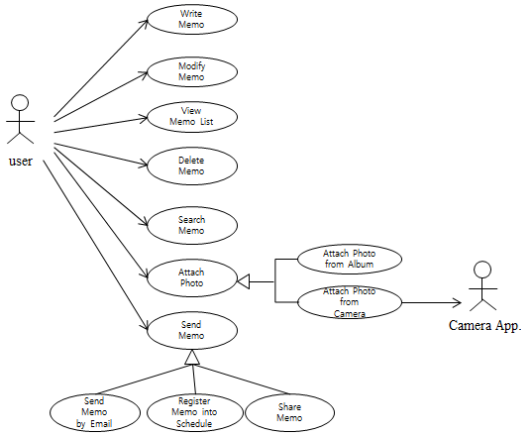
이 장에서는 본 논문에서 제시하는 메타모델들을 기반으로 실제 앱 개발 프로세스에 적용하여 생성된 모델들에 대한 사례 연구를 제시하고자 한다.

4.1 사례 연구

본 논문에서는 ‘메모’ 앱을 모델링 사례 연구로 채택하였다[11-14].

4.1.1 유스 케이스 기반의 요구사항 식별

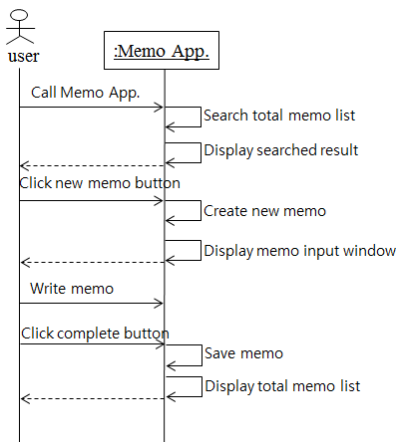
Fig. 4의 메타모델을 적용하여 메모 앱의 기능 요구사항들을 식별하여 Fig. 9와 같이 유스 케이스 다이어그램으로 표현하였다.



[Fig. 9] Use Case Model

4.1.2 시스템 상호작용 모델링

식별된 기능 요구사항들에 대한 각각의 유스 케이스에 대해 각각의 유스 케이스 내의 사건 흐름을 모델링하기 위해 Fig. 5의 메타 모델을 적용하여 Fig. 10과 같이 시스템 순차도를 작성하여 시스템과 사용자와의 상호작용을 모델링 하였다.

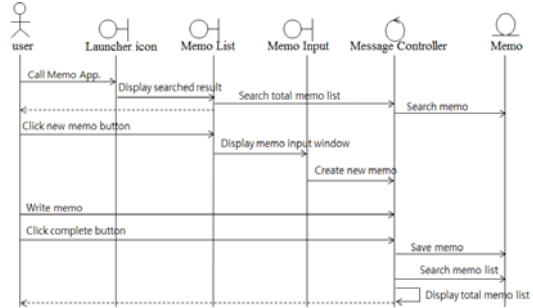


[Fig. 10] System Sequence Diagram

4.1.4 MVC 기반 상호 작용 모델링

작성된 시스템 순서도를 입력물로 받아서 앱 내에 참

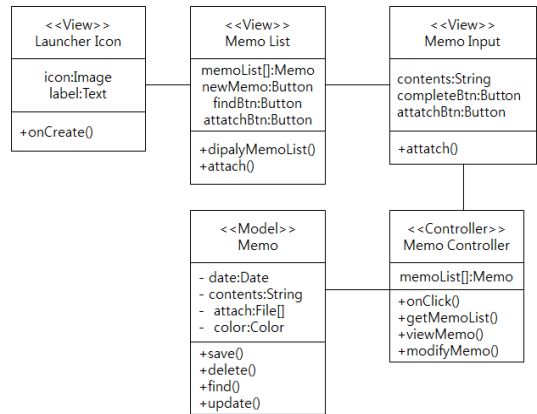
여하는 객체들을 MVC 패턴을 적용하여 3가지 형태의 클래스(또는 객체)들로 분류하여 UML의 상호 작용도를 이용하여 객체 간의 상호작용을 설계하였다. Fig. 11은 Fig. 6의 메타 모델을 적용하여 생성된 결과이다.



[Fig. 11] MVC-based Interaction Diagram

4.1.5 MVC 기반 클래스 설계

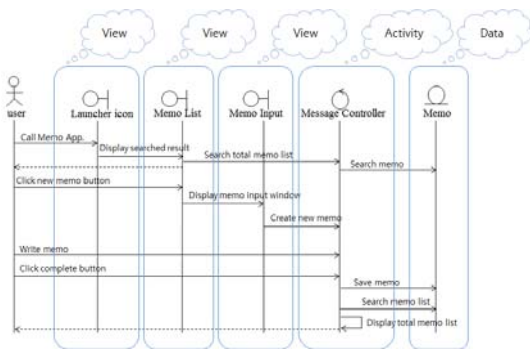
Fig. 11의 상호 작용도를 입력물로 받아서 클래스들을 정의하고, 클래스에 필요한 속성, 메소드를 정의하고, 클래스들 간의 관계를 Fig. 7의 메타모델을 적용하여 설계 하였다.



[Fig. 12] MVC-based Class Diagram

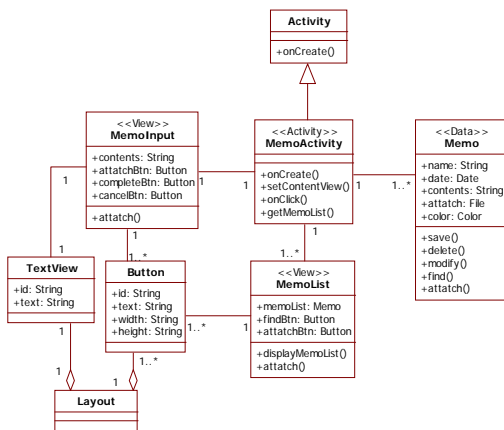
4.1.4 PSM 클래스 설계

Fig. 11의 상호 작용도와 Fig. 12의 MVC 기반 클래스 다이어그램을 입력물로 받아서 안드로이드 플랫폼의 특징들을 반영하여 클래스 다이어그램을 정제하면 Fig. 13과 같다.



[Fig. 13] Refining Classes by Interaction Diagram

상호 작용도에 표현된 바운더리 클래스들은 뷰 클래스로 매핑되고, 컨트롤러 클래스는 Activity 클래스로 매핑되고, 엔티티 클래스는 Data 클래스로 매핑된다. 이를 바탕으로 Fig. 8의 메타모델을 적용하여 정제된 PSM 클래스 모델은 Fig. 14와 같다.



[Fig. 14] PSM Class Model

5. 결론 및 향후 연구과제

모바일 앱 개발에 대한 공급이 증가하다 보니 체계적인 개발 프로세스나 모델없이 후후죽순처럼 개발되는 상황이 이루어지고 있다. 이런 환경 하에서 개발된 앱들은 독립성, 재사용성, 확장성 등과 같은 품질 요소들이 고려되어 있지 않기 때문에 품질이 저하되어 사용자들로 하여금 앱에 대한 만족도가 매우 저평가될 위험이 높다. 본 논문에서는 이러한 문제점을 해결하고자 모바일 앱 개발의 생산성을 향상시키면서 앱 개발에 있어서 일관성 있

으면서 체계적이고 편리하게 개발할 수 있도록 하기 위한 개발 프로세스를 메타모델로 설계하였다.

본 논문에서 모바일 앱 개발 프로세스를 메타모델로 제시한 이유는 여러 다양한 형태의 앱 개발 프로젝트에 있어서 해당 앱의 목적에 맞게 커스터마이징할 수 있도록 하기 위함이다. 특히 본 논문에서는 안드로이드 기반의 모바일 앱 개발에 맞춰서 MVC 패턴과 객체지향 기법을 접목시킨 메타모델을 제시하였다. 이는 모델링에서부터 MVC 패턴을 적용함으로써 앱 구현단계에 있어서 보다 자동 코드 생성으로의 구현이 가능하게 된다. 이는 결국 안드로이드 프로그램 자동 생성 메커니즘을 위한 기반이 되는 것이다. 따라서 향후 연구 과제로는 제시한 메타모델을 기반으로 자동화 도구를 구축하는 것이다.

References

- [1] SA(Strategy Analytics), SA Report, <http://blogs.strategyanalytics.com/>
- [2] Apple ios-7, <http://www.apple.com>
- [3] Google Android [Online]. <http://www.android.com>
- [4] Salmre, I., Writing Mobile Code: Essential Software Engineering for Building Mobile Applications, Addison-Wesley Professional, 2005.
- [5] S. Kim, Android Programming Complete Guide, Hanbit-Media, 2011.
- [6] C. Y. Song, E. S. Cho, and C. J. Kim. A user interface hierarchical modeling process based on a metamodel. Journal of Korea Multimedia Society 11(4)(2008), pp.525-543.
- [7] Object Management Group. MDA Documents. <http://www.omg.org/docs/omg/03-06-01.pdf>,2003.
- [8] Erich Gamma et. al, Design Patterns-Elements of Reusable Object-Oriented Software, Addison-Wesley, 2012.
- [9] Craig Larman, Applying UML and Patterns, Prentice Hall, 2001.
- [10] Martin Fowler, UML Distilled, Addison-Wesley,2003.
- [11] Eun Sook Cho, Chul Jin Kim, and Sook Hee Lee, "A Modeling Technique for Development of Mobile App. based on Android", Vol.14, No.8, pp.3999-4005. 2013.
- [12] B.-H. Oh, K.-S. Hong, "Multi-Modal User Distance Estimation System based on Mobile Device", The Journal of The Institute of Internet, Broadcasting and Communication, Vol. 14, No. 2, pp. 65-71, Apr. 2014.
- [13] K. Ro, H. Hwang, S. Kim, "A Research on Personalized

Mobile Advertising Service using the Linkage between Digital Signage and Smartphones”, The Journal of The Institute of Internet, Broadcasting and Communication, Vol. 14, No. 1, pp. 139-146, Feb. 2014.

- [14] H.-T. Seok, "Hybrid Web App Development for Eye movement at Mobile Devices”, The Journal of The Institute of Internet, Broadcasting and Communication, Vol. 13, No. 6, pp. 249-254, Dec. 2013.

조 은 숙(Eun-Sook Cho)

[정회원]



- 1993년 2월 : 동의대학교 전산통계학과 졸업(이학사)
- 1996년 2월 : 숭실대학교 대학원 컴퓨터학과(공학석사)
- 2000년 2월 : 숭실대학교 대학원 컴퓨터학과(공학박사)
- 2000년 9월 ~ 2005년 2월 : 동덕여자대학교 정보학부 강의전임교수
- 2005년 3월 ~ 현재 : 서일대학교 컴퓨터 소프트웨어과 부교수

<관심분야>

Software Engineering, Embedded Software, Mobile Computing