

안드로이드 기반의 모바일 앱 개발을 위한 모델링 기법

조은숙^{1*}, 김철진², 이숙희³

¹서일대학교 컴퓨터소프트웨어과, ²인하공업전문대학교 컴퓨터시스템과

³서경대학교 컴퓨터과학과

A Modeling Technique for Development of Mobile App. based on Android

Eun-Sook Cho^{1*}, Chul-Jin Kim² and Sook-Hee Lee³

¹Dept. of Computer Software, Seoil University

²Dept of Computer System, Inha Technical College

³Dept of Computer Science, Seokyeong University

요 약 스마트 폰 2천만 시대가 도래 하면서 향후 모바일 시장의 규모는 엄청나게 거대해질 것으로 전망되고 있다. 이러한 흐름에 동반하여 소프트웨어 개발의 형태도 웹 기반 소프트웨어에서 모바일 기반의 앱 형태나 클라우드 컴퓨팅 기반의 서비스 형태로 변화되고 있다. 현재 전 세계적으로 스마트폰을 위한 플랫폼으로는 구글의 안드로이드와 애플의 IOS가 양대 산맥을 이루고 있다. 이러한 플랫폼 하에서 실행되는 모바일 앱을 개발할 경우 기존의 소프트웨어 모델링 기법을 그대로 적용할 수가 없다. 따라서 본 연구에서는 기존의 소프트웨어 모델링 기법을 안드로이드 기반의 소프트웨어 개발에 맞추어 특화된 모델링 기법을 제안한다. 안드로이드 기반의 모바일 스마트 환경에 필요한 분석 및 설계 기법을 모델링 기법으로 제안한다.

Abstract We forecast that the scale of mobile market will be large scale according to the twenty million times of smart phone's user is arrived. The type of software development is being transferred into mobile-based App. or cloud computing-based service from web-based software. Currently both Google's android and Apple's IOS are known for two prominent platforms. Existing modeling techniques cannot be applied into the case of mobile application which is executed those platforms. Therefore, we suggest a modeling technique that is customized existing modeling techniques into android-based software development. That is, we propose an analysis and design techniques which are needed into android-based mobile smart environment.

Key Words : Mobile-based App., Cloud Computing, Modeling Technique, Android

1. 서론

2012년 가트너 보고[1]에 따르면 세계 휴대폰 시장은 스마트폰 시장의 견조한 성장으로 2,828억 불 규모로 10.5% 성장을 예상하였고, 사상 처음으로 시장규모 6억 대에 진입할 것으로 예측하였다. 이러한 흐름에 동반하여 소프트웨어 개발의 형태도 웹 기반 소프트웨어에서 모바일

기반의 앱 형태나 클라우드 컴퓨팅 기반의 서비스 형태로 변화되고 있다. 현재 전 세계적으로 스마트폰을 위한 플랫폼으로는 구글의 안드로이드[2]와 애플의 IOS[3]가 양대 산맥을 이루고 있다. 이처럼 모바일 앱에 대한 개발과 사용이 급속도로 증가하면서 모바일 앱의 품질에 대한 요구사항이 증가되고 있다. 특히 유료 모바일 앱의 경우는 더욱 그러한 실정이다. 이러한 요구사항을 해결하

본 논문은 2012년 서일대학교 교내연구과제로 수행되었음.

*Corresponding Author : Chul-Jin Kim(Inha Technical College)

Tel: +82-32-870-2338 email: cjkim777@gmail.com

Received July 23, 2013

Revised August 1, 2013

Accepted August 7, 2013

기 위해서는 모바일 앱 개발에 있어서 모바일 앱 특성에 최적화 된 모델링 기법이 반영되어야 한다. 테스트 기법 등 다양한 기법들이 이러한 플랫폼 하에서 실행되는 모바일 앱을 개발할 경우 기존의 소프트웨어 모델링 기법을 그대로 적용할 수가 없다. 그래도 적용할 경우, 모바일 앱이 갖는 특성이 제대로 반영이 될 수 없기 때문에 앱의 품질이 저하 될 수 있다. 따라서 본 연구에서는 기존의 소프트웨어 모델링 기법을 안드로이드 기반의 소프트웨어 개발에 맞도록 새롭게 개발하였다.

본 논문의 구성으로 2장에서는 관련연구로 안드로이드 모바일 플랫폼과 모바일 앱 개발 기법에 적용할 MVC 패턴과 UML에 대해 설명하며, 3장에서는 본 논문에서 제안하는 안드로이드 기반의 모바일 앱 개발 프로세스와 모델링 기법을 제시한다. 4장에서는 3장에서 제안한 기법을 실제 사례에 적용한 결과를 제시한다. 마지막으로 5장에서 결론 및 향후 연구 과제를 제시한다.

2. 관련 연구

2.1 안드로이드 플랫폼

안드로이드는 구글에서 만든 모바일 기기를 위한 플랫폼으로서, 운영체제, 미들웨어 및 주요 응용 프로그램을 포함하는 모바일 기기용 소프트웨어 모음을 의미한다[2]. 안드로이드는 다음과 같은 기술적 특징들을 지니고 있다.

■ 개방성

안드로이드는 개발자들이 핸드셋을 이용해 대단한 모바일 애플리케이션을 만들어 내었던 일들이 바탕이 되어 개발되었다. 안드로이드는 개방형 리눅스 커널을 기반으로 하고 있다. 나아가, 그것은 메모리와 하드웨어 리소스를 모바일 환경에 최적화시킨 맞춤형 버추얼(가상) 기기를 사용하고 있다.

■ 다양한 애플리케이션의 동등성

안드로이드는 전기기의 기본 애플리케이션과 부가 애플리케이션을 구별하지 않는다. 다양한 종류의 애플리케이션과 서비스를 사용자에게 제공하는 전기기의 성능에 동일한 접속이 가능하도록 설계되었다. 안드로이드 플랫폼 기반의 기기로서, 사용자들은 자신의 기호에 따라 전화기를 맞춤화 할 수 있다.

■ 상호 호환 및 연동성

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.

■ 애플리케이션 개발의 용이성

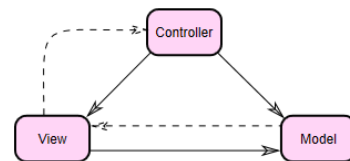
안드로이드는 풍부한 애플리케이션 설계를 위한 다양하고 유용한 라이브러리와 툴을 제공할 뿐만 아니라 개발자들에게 높은 생산성과 그들의 기기에 대한 깊은 통찰력을 주는 플랫폼을 기반으로 개발된 통합 툴세트를 포함하고 있다.

2.2 MVC 패턴

MVC(Model View Controller)는 소프트웨어 공학에서 사용되는 아키텍처 패턴이다[4,5].

여기서 모델(Model)은 데이터구조를 표현한다. 일반적으로 모델 클래스는 데이터를 추출,입력,갱신하는 등의 함수를 포함하게 된다.

그림 1에 표현된 것처럼 뷰(View)는 사용자에게 보여질 부분을 표현한다. 예를 들면 텍스트, 체크박스 항목 등과 같은 사용자 인터페이스 요소나 웹 페이지 등으로 사용된다. 때로는 헤더(Header)나 푸터(Footer)처럼 웹페이지의 조각일 수도 있다. 또한 RSS 페이지나, 혹은 다른 페이지 형태의 것일 수 있다.



[Fig. 1] MVC Pattern

컨트롤러(Controller)는 일반적으로 모델과 뷰 사이에서 동작한다.

이 패턴을 성공적으로 사용하면, 사용자 인터페이스로부터 비즈니스 로직을 분리하여 애플리케이션의 시각적 요소나 그 이면에서 실행되는 비즈니스 로직을 서로 영향 없이 쉽게 고칠 수 있는 애플리케이션을 만들 수 있다. 안드로이드 기반의 모바일 프로그래밍 패턴을 보면 이 MVC 패턴을 기반으로 하고 있다. 사용자 인터페이스 역할을 나타내는 뷰는 xml로 정의하고, 비즈니스 역할을 하는 컨트롤러는 자바 클래스로 작성하고, 데이터 역할을 하는 모델은 데이터 부분을 갖는 별도의 xml 파일로 정의하고 있다.

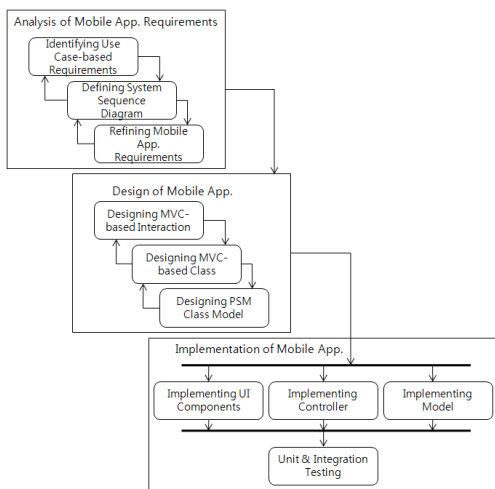
3. 모바일 앱 개발 프로세스와 모델링 기법

이 장에서는 안드로이드 기반의 모바일 앱 개발을 위한 개발 프로세스를 정의하고, 개발 프로세스 가운데 앱 개발에 필요한 모델링 기법을 제시하고자 한다.

3.1 모바일 앱 개발 프로세스

현재 산업체에서 많이 쓰이고 있는 개발 프로세스 형태는 반복적이면서 점진적(Incremental)으로 개발하는 형태이다. 그 이유는 이러한 개발 프로세스가 개발 기간 동안의 위험(Risk)을 줄일 수 있을 뿐만 아니라 개발 기간의 단축 및 품질 개선의 효과를 가져올 수 있기 때문이다. 이러한 개발 프로세스의 대표적인 형태가 UP(Unified Process)이다[6,7].

본 연구에서는 이러한 프로세스를 기반으로 한 모바일 앱 개발 프로세스를 다음과 같이 정의한다.



[Fig. 2] Development Process for Mobile App.

모바일 앱 개발 프로세스는 크게 4개의 단계인 모바일 앱 요구사항 분석, 모바일 앱 컴포넌트 설계, 모바일 앱 설계, 모바일 앱 구현 단계로 나뉘어 진다.

3.2 UML 기반의 모바일 앱 모델링 기법

안드로이드 플랫폼의 아키텍처를 살펴보면 내부 구성 요소들이 컴포넌트 또는 클래스들로 구성되어있다. 그리고 계층화 아키텍처(Layered Architecture) 스타일을 기반으로 하고 있다. 따라서 본 본문에서는 이러한 패러다임을 모바일 앱 설계에 반영하기 위해서 객체지향 모델링

언어인 UML을 적용하고자 한다[8-16]. 본 논문에서 제시하는 모바일 앱 모델링 기법은 앞서 정의한 모델링 객 개발 프로세스의 2가지 단계인 모바일 앱 컴포넌트 설계와 모바일 앱 설계 단계에 적용된다.

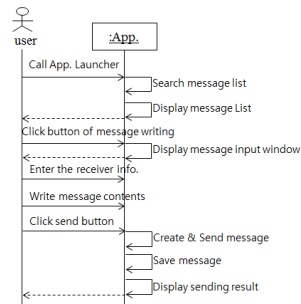
3.2.1 유스 케이스기반의 모바일 앱 기능 요구사항 식별

모바일 앱은 사용자와 모바일 디바이스 간의 상호작용을 수반하게 된다. 따라서 개발하고자 하는 모바일 앱의 기능들을 원활하게 식별하기 위해서는 UML의 유스 케이스 모델을 적용하여 추출한다.

한 개의 유스 케이스가 해당 앱의 한 개의 독립적인 기능으로 추출되며, 해당 유스 케이스와 상호작용할 외부 개체를 액터(Actor)로 추출하여 표현한다.

3.2.2 시스템 순서도 작성

작성된 유스 케이스 모델을 가지고 각각의 유스 케이스 별로 해당 유스 케이스의 작업 흐름을 표현하기 위해 시스템 순서도를 작성한다. 이 다이어그램은 UML의 순차도(Sequence Diagram)를 응용한 것으로서, 유스 케이스 모델의 유스 케이스를 시스템(System) 객체로 표현하고, 액터와 해당 유스 케이스와 어떤 순서로 상호작용하는지를 표현하는 다이어그램이다[그림 3 참조].

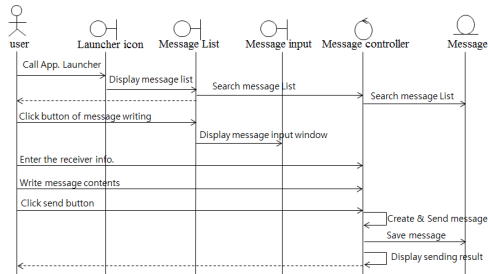


[Fig. 3] System Sequence Diagram

3.2.3 MVC 기반 상호 작용(Interaction) 설계

앞서 작성된 시스템 순차도에는 객체가 '앱(App.)' 하나로 모두 표현된다. 여기서 이 앱이란 것은 개발될 모바일 앱 전체를 의미한다. 그러나 각각의 기능 별로 해당 앱에는 여러 개의 화면과 여러 개의 객체들이 서로 상호 작용하며 기능들을 수행하게 된다. 따라서 총체적인 단위로 표현된 앱 객체를 보다 세분화 시켜서 표현해야 해당 기능 별로 필요한 컴포넌트나 객체들을 추출할 수 있다. 그러기 위해서여기서는 앞서 작성한 시스템 순차도의 시스템 객체가 수행한 메시지들을 그 메시지의 성격에 따

라 MVC 패턴을 적용하여 사용자 인터페이스(User Interface) 객체(뷰객체, 바운더리 객체), 컨트롤러 객체, 데이터 객체(모델 객체) 등으로 분할하여 메시지들을 해당 객체들로 할당한다. 이를 객체 또는 컴포넌트 간 상호작용도(Interaction Diagram)라 한다.

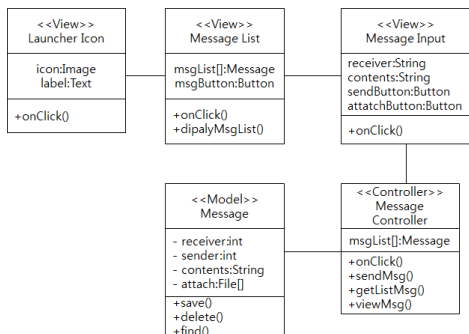


[Fig. 4] MVC-based Interaction Diagram

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 UML의 스테레오타입을 이용하여 명시한다.

예를 들어 그림 4의 상호 작용도에서는 총 4개의 클래스가 추출되며, Launcher icon과 Message Input 클래스는 뷰 클래스가 되고, Message Controller 클래스는 Controller 클래스가 되고, Message 클래스는 Model 클래스가 된다. 이를 UML의 클래스 다이어그램으로 설계하면 그림 5와 같다.



[Fig. 5] MVC-based Class Model

3.2.5 PSM(Platform Specific Model) 클래스 모델 생성

이전 과정들은 플랫폼에 독립적인 PIM(Platform Independent Model)을 생성하는데 관련된 기법들을 제시

하였다. 여기서는 MVC 기반으로 클래스 모델이 설계되면 이를 입력물로 받아서 안드로이드 플랫폼의 패턴을 반영한 PSM 클래스 모델을 작성한다. 이렇게 함으로써 코드 자동 생성 뿐만 아니라 특정 플랫폼에 최적화된 모바일 앱 구현을 용이하게 해 준다. PIM 모델을 안드로이드에 종속적인 PSM 모델로 변환하기 위해 필요한 원칙은 다음과 같다.

- 원칙1) 사용자와 상호 작용하는 뷰 클래스는 Activity 클래스가 된다.
- 원칙2) 서로 다른 뷰 클래스는 하나의 Activity 클래스로 대응된다.
- 원칙3) 첫 실행 뷰 클래스는 다음 뷰 클래스와 함께 하나의 Activity로 대응된다.
- 원칙4) Controller 클래스의 경우 화면에 보이지 않는 Service 클래스가 된다.
- 원칙5) 모델 클래스는 일반 Java Class로 대응된다.
- 원칙6) 뷰 클래스 간에 전송되는 정보가 있을 경우 이것은 Intent로 대응된다.

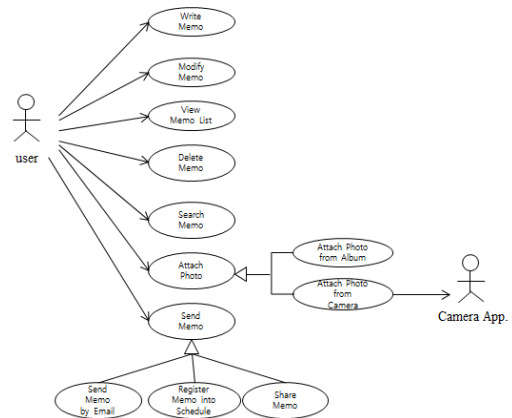
4. 평가

이 장에서는 본 논문에서 제시하는 모델링 기법을 적용한 사례 연구를 제시하고자 한다.

본 논문에서는 ‘메모’ 앱을 모델링 사례 연구로 채택하였다.

4.1 유스 케이스 기반의 기능요구사항 식별

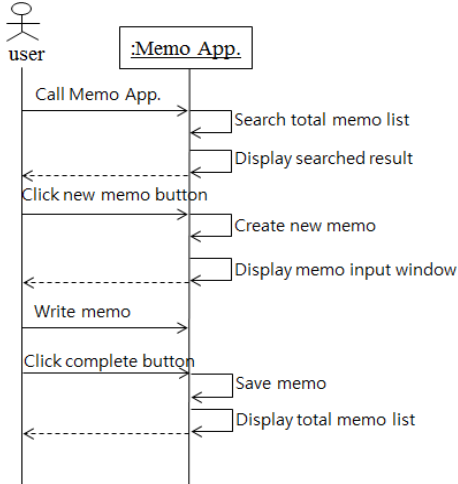
메모 앱의 기능 요구사항을 유스 케이스로 식별한 유형은 그림 6과 같다.



[Fig. 6] Use Case Model

4.2 시스템 순서도 작성

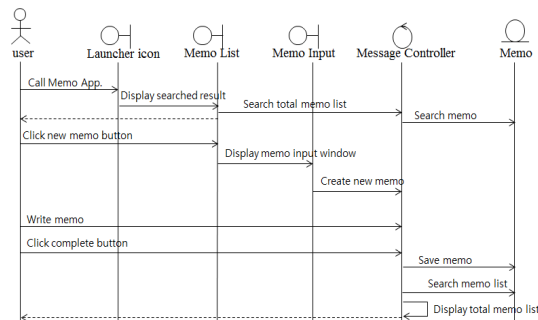
추출된 유스케이스 별로 각각의 유스케이스 내의 사건 흐름을 모델링하기 위해 시스템 순서도를 작성하였다.



[Fig. 7] System Sequence Diagram

4.3 MVC 기반 상호 작용 설계

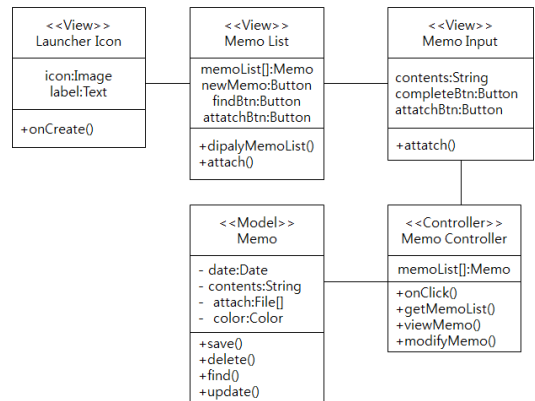
작성된 시스템 순서도를 기반으로 앱 내에 가담하는 객체들을 MVC 패턴을 기반으로 분류하여 UML의 상호 작용도를 이용하여 객체 간의 상호작용을 설계하였다.



[Fig. 8] MVC-based Interaction Diagram

4.4 MVC 기반 클래스 설계

작성된 상호 작용도를 입력물로 받아서 클래스들을 추출하고, 클래스에 필요한 속성, 메소드를 정의하고, 클래스들 간의 관계를 설계하였다.



[Fig. 9] MVC-based Class Diagram

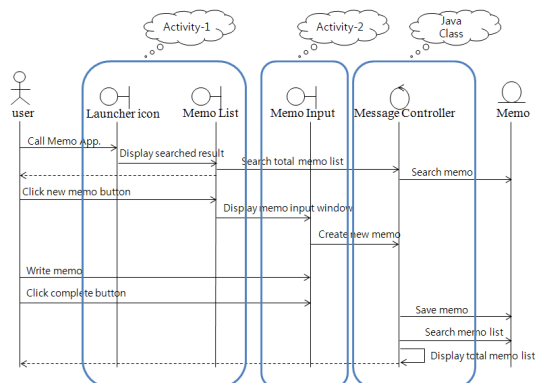
4.5 PSM 클래스 모델 생성

MVC 기반의 클래스 다이어그램은 플랫폼에 독립적인 모델이다. 그러나 설계 이후 구현으로의 자동 코드 생성이 실현되기 위해서는 플랫폼에 종속적인 PSM 모델이 필요하다. 본 논문에서는 위의 클래스 다이어그램을 안드로이드 플랫폼에 종속적인 클래스 다이어그램으로 정제화 하였다. PIM(Paltform Independent Model)에서 PSM으로 변환하는 과정에 적용되는 변환 규칙은 다음과 같다.

[Table 1] Mapping Rule from PIM to PSM

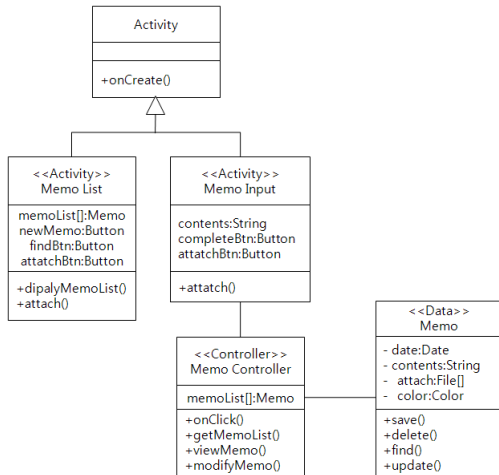
PIM	PSM
View	Activity
Controller	Controller
Model	Data
Interface	Intent

상호 작용도를 통해 클래스를 다시 정제하면 그림 10과 같다.



[Fig. 10] Refining Classes by Interaction Diagram

상호 작용도에 표현된 뷰 클래스들 가운데 어떤 뷰 클래스는 한 개의 뷰 클래스가 하나의 Activity가 되고, 어떤 경우는 하나 이상의 뷰 클래스가 하나의 Activity가 되는 경우도 존재한다. 이를 바탕으로 정제된 PSM 클래스 모델은 그림 11과 같다.



[Fig. 11] PSM Class Model

5. 결론 및 향후 연구과제

스마트 폰 사용자가 급속도로 증가함으로써 모바일 앱 시장 또한 이에 맞춰 증가하게 되었다. 그러다보니 모바일 앱 개발에 대한 수요나 공급이 증가하게 되었다. 그런데 모바일 앱은 기존의 소프트웨어와는 플랫폼이 다른 환경에서 작동되기 때문에 모바일 앱 개발에 있어서 모델링 기법 또한 기존의 기법과 다른 접근이 필요한 것이다. 본 논문에서는 모바일 앱 개발의 생산성을 높일 뿐만 아니라 앱 개발에 있어서 일관성 있으면서 편리하게 개발할 수 있도록 하기 위해 컴팩트한 모델링 기법을 제시하였다. 특히 본 논문에서 제시한 기법은 안드로이드 플랫폼 상에서 작동할 모바일 앱 개발에 필요한 모델링 기법을 제시하기 위해 MVC 패턴과 객체지향 기법을 접목시켰다. MVC 패턴을 적용한 이유는 안드로이드 기반 구현 패턴이 MVC 패턴을 따르고 있기 때문이다. 따라서 모델링에서부터 MVC 패턴을 적용하면 자연스럽게 구현 단계에 적용이 용이하게 된다. 이는 결국 안드로이드 프로그램 자동 생성 메커니즘을 위한 기반이 되는 것이다. 따라서 향후 연구 과제로는 제시한 기법을 보다 체계적이면서 정확화 된 형태로 정제하여 안드로이드 기반 모바일 앱 개발 자동화 도구를 개발하는 것이다.

References

- [1] Gartner, Gartner Report, <http://www.gartner.com>
- [2] Google Android [Online]. <http://www.android.com>
- [3] Apple ios-7, <http://www.apple.com>
- [4] Erich Gamma et. al, Design Patterns-Elements of Reusable Object-Oriented Software, Addison-Wesley, 2012.
- [5] Craig Larman, Applying UML and Patterns, Prentice Hall, 2001.
- [6] Martin Fowler, UML Distilled, Addison-Wesley, 2003.
- [7] John Cheesman, John Daniels, UML Components, Addison-Wesley, 2013.
- [8] Salmre, I., Writing Mobile Code: Essential Software Engineering for Building Mobile Applications, Addison-Wesley Professional, 2005.
- [9] S. Kim, Android Programming Complete Guide, Hanbit-Media, 2011.
- [10] Kim C. J., Cho E. S., Song C. Y., "A Design Technique of Configurable Framework for Home Network Systems", Journal of the Korea Academia-Industrial Cooperation Society, Vol. 12, No.4, pp.1844-866, April 2011.
DOI: <http://dx.doi.org/10.5762/KAIS.2011.12.4.1844>
- [11] Eun Sook Cho, Chul Jin Kim, "A Technique of Applying Ontology for Service Customization of Android", Journal of the Korea Academia-Industrial Cooperation Society, Vol 13, No.6, pp.2707-2712, June 2012.
DOI: <http://dx.doi.org/10.5762/KAIS.2012.13.6.2707>
- [12] S.-C. Lim, "A Study of Android Launcher based on Application Virtualization", Journal of The Institute of Webcasting, Internet and Telecommunication, Vol 13, No 2, pp. 115~120, 2013.
- [13] K. Hwang, "Video Player for Online SVC Stream in Android Platform", Journal of The Institute of Webcasting, Internet and Telecommunication, Vol 12, No 1, pp. 157~164, 2012.
- [14] J.-y. Jang, I. Kim, "Developing a Schedulable App for Switching to Silent Mode Using an Android phone", Journal of The Institute of Webcasting, Internet and Telecommunication, Vol 12, No 2, pp. 53~59, 2012.
- [15] S.-J. Oh, "Design of a Middleware for Android-based Smart Phone Applications", Journal of The Institute of Webcasting, Internet and Telecommunication, Vol 12, No 2, pp. 111~117, 2012.
- [16] J. Oh, B. An, "Android Based Mobile Smart Alarm", Journal of The Institute of Webcasting, Internet and Telecommunication, Vol 12, No 3, pp. 141~148, 2012.

조 은 숙(Eun-Sook Cho)

[정회원]



- 1993년 2월 : 동의대학교 전산통계학과 (이학사)
- 1996년 2월 : 송실대학교 대학원 컴퓨터학과 (공학석사)
- 2000년 2월 : 송실대학교 대학원 컴퓨터학과 (공학박사)
- 2000년 9월 ~ 2005년 2월 : 동덕여자대학교 강의전임교수
- 2005년 3월 ~ 현재 : 서일대학교 컴퓨터소프트웨어과 부교수

<관심분야>

Software Engineering, Embedded Software, Mobile Computing

김 철 진(Chul-Jin Kim)

[정회원]



- 1996년 2월 : 경기대학교 전자계산학과 (학사)
- 1998년 2월 : 송실대학교 대학원 컴퓨터학과 (공학석사)
- 2004년 2월 : 송실대학교 대학원 컴퓨터학과 (공학박사)
- 2004년 9월 ~ 2009년 2월 : 삼성전자 책임연구원
- 2009년 3월 ~ 현재 : 인하공전 컴퓨터시스템과 조교수

<관심분야>

Software Engineering, Embedded Software, Mobile Computing

이 숙 희(Sook-Hee Lee)

[정회원]



- 1979년 2월 : 숙명여자대학교 독문학과 졸업 (학사)
- 1982년 2월 : 동국대학교 경영대학원 정보처리학과 졸업 (석사)
- 1991년 2월 : 성균관대학교 대학원 통계학과 졸업 (박사)
- 1987년 3월 ~ 1993년 2월 : 동신대학교 전자계산학과 교수
- 1993년 3월 ~ 현재 : 서경대학교 컴퓨터과학과 교수

<관심분야>

Software Engineering, Software Testing, OOSE