

MDA 기반의 모바일 크로스 프레임워크 설계

송유진[†], 이은주^{**}, 한덕수^{***}

Design Mobile Cross Framework Based MDA

Yujin Song[†], Eun-Joo Lee^{**}, Deok-Soo Han^{***}

ABSTRACT

Recently Mobile Software Applications are developed in various languages and stored in App Store. App Users selectively use appropriate apps for the owned hardware. In other words, it depends on the kinds of operating systems of the apps whether to use or not to use the applications in various languages. The apps should be differently implemented according to the kind of the user's device, though they provide the same functions. To solve these problems, it is necessary to define an independent function specification method which is not dependent to a specific system environment. In this paper, the Mobile Application Developing Framework is suggested, which incorporates all of the development process. Standardized models are proposed which can be used in the analysis and design steps. In implementation phase, a technique for cross framework design is suggested so as to implement a platform dependent mobile app.

Key words: Component Based Development, Model Driven Architecture, Mobile Application Logic Analysis

1. 서 론

최근 모바일 소프트웨어 개발환경은 스마트기기 활용도 증가 추세에 따라 급속히 변화하고 있다. 모바일 소프트웨어 어플리케이션(이하 앱과 동일한 의미로 사용됨)은 운영체제에 따라 다양한 언어로 개발되어 앱 스토어에 저장된 후 앱 사용자는 보유한 하드웨어 기종에 맞는 앱을 선택하여 사용한다. 즉, 현재 개발되고 있는 어플리케이션들은 다양한 운영체제에 적합하도록 다양한 언어로 개발되어 운영체제 종류에 따라 선택되어 사용된다. 이는 동일한 기능을 수행하는 어플리케이션일지라도 어플리케이션이 사용될 디바이스에 따라 각각 개발을 달리

해야한다는 문제점이 있고, 소프트웨어공학 측면에서 재사용성과 유지보수성 그리고 이식성이 떨어진다는 것을 의미한다.

이와 같은 문제점을 해결하기 위해 특정 기능을 수행하는 어플리케이션을 개발 초기 설계단계부터 플랫폼에 비종속적인 설계모델을 적용하고, 그 어플리케이션의 메타모델을 저장소에 저장된 후 플랫폼에 종속적인 설계모델로 변환하여 재사용할 수 있게 한다면 개발비용과 유지보수 비용을 절감할 수 있을 것이다. 이를 위해서는 첫째, 특정 시스템 환경에 종속되지 않는 독립적인 기능 명세방법이 필요하고, 둘째, 그 앱을 개발하는 표준화된 모바일 소프트웨어 개발 프로세스가 필요하다. 최근 모바일 앱 개발

※ Corresponding Author : Deok-Soo Han, Address: Changha-ri, Gogyeong-myeon, Yeongcheon-si, Gyung-sang-buk-do, Republic of Korea (38900), TEL : +82-54-330-4740, FAX : +82-54-335-5790, E-mail : dshan123@hanmail.net

Receipt date : Jul. 28, 2016, Approval date : Aug. 3, 2016

[†] School of Computer Science and Engineering Kyungpook National University
(E-mail : yujinkor@hanmail.net)

^{**} School of Computer Science and Engineering Kyungpook National University
(E-mail : ejlee@knu.ac.kr)

^{***} Dept. of Computer Eng. Korea Army Academy at Yeong-Cheon

연구 중에서 모바일 앱의 메타모델 기반 프레임워크 설계 연구는 다양한 형태의 앱 특성을 고려한 분류와 목적에 맞는 목표지향성은 있으나 안드로이드 운영체제 플랫폼에 의존적인 개발방법이고 서로 다른 플랫폼으로 모바일 어플리케이션을 개발하기 위해서는 설계 초기 단계부터 반복 작업이 발생한다는 단점이 있다[1].

본 논문에서 제안하는 모바일 앱 개발 프레임워크는 앱 설계에서 개발까지의 전 과정을 세분화하여, 분석과 설계단계에서 사용되는 표준 개발 모델을 제시하고 구현단계에서 플랫폼에 종속적인 모바일 앱으로 구현할 수 있는 크로스 프레임워크 설계 방법을 제안한다.

논문의 구성은 2장에서 본 논문의 기본 배경이 되는 MDA(Model Driven Architecture)와 Cross-Platform의 정의와 안드로이드 환경의 모바일 아키텍처에 대해 설명하고, 3장에서는 제안하는 MDA를 이용한 크로스 프레임워크의 전체적인 구조와 MDA를 이용한 개발 프로세스에 의해 서술한다. 4장에서는 제안한 프로세스로 적용 사례를 제시하며 5장에서 결론을 맺는다.

2. 연구배경

2.1 MDA(Model Driven Architecture)

OMG(the Object Management Group)는 시스템을 명세하고 새로운 방법으로 시스템을 구축하기 위한 표준안으로 MDA를 제시하였다. MDA는 상호운영성과 시스템 통합에 있어서 모델중심의 개발 방법으로 UML(Unified Modeling Language), MOF(Meta Object Facility), CWM(Common Warehouse Metamodel), XML(XML Metadata Interchange)등 모델링 표준화 작업의 결과물로 설계수준부터 구현까지의 표준 모델을 제시하고 있다. 특히 미들웨어인 CORBA(Common Object Request Broker Architecture)를 이용하여 실시간 시스템과 임베디드 시스템 등 많은 분야로 활용 영역을 넓이고 있다. MDA는 시스템 설계와 명세화를 구현하고자 하는 플랫폼과 분리하여 시스템을 보다 효율적으로 관리하고 유지할 수 있는 장점을 갖는다.

따라서 MDA방식을 적용한 시스템은 기술 변화에 원활한 대처능력을 가지며 플랫폼 간에 이식성이

높다. 이렇게 개발된 시스템은 유지 보수 비용이 적고, 개발되는 시스템의 생명주기도 길어진다. MDA 구조에서는 우선 PIM(Platform Independent Model)과 PSM(Platform Specification Model)으로 모델을 구분하고 모두 UML로 정의한다. PIM은 특정 플랫폼에 독립적인 방법으로 시스템을 설계한 모델을 말하며, 결과물로 PIM에 정형화된 변환 법칙을 이용하여 PSM을 생성한다.

따라서 MDA기반 개발 절차는 사용자 요구사항을 파악하고 특정 플랫폼에 비종속적인 도메인에 대한 기능적 분석과 설계를 UML로 작성한다. 그 다음 특정 플랫폼에 종속적인 도메인의 기능을 구현하고자 하는 어플리케이션과 매핑하여 기술하고 구현을 위한 소스코드 생성작업 과정을 거치게 된다. 이러한 과정을 통하여 기술 플랫폼에 대한 신속한 대응력도 향상되고 개발생산성도 높아져 양질의 시스템을 개발할 수 있는 밑거름이 된다[2].

2.2 Cross-Platform

크로스 플랫폼은 여러 종류의 컴퓨터 플랫폼에서 동작할 수 있다는 의미로 최근에는 멀티플랫폼 소프트웨어라고도 말한다. 현재 크로스 플랫폼을 지원하는 소프트웨어로는 자바, 오페라, 파이어폭스 등이 있다. 다양한 언어의 각 플랫폼에 동일하게 적용할 수 있는 어플리케이션을 개발할 수 있도록 하려면 먼저 다양한 플랫폼의 공통사항을 찾아내야 한다. 현재 개발되고 있는 어플리케이션 개발방식을 살펴보면 아이폰 OS의 경우 개발언어는 Objective-C이고, 안드로이드 OS의 개발언어는 Java, 윈도우즈 Phone7의 OS는 C#이며, 국내기업인 삼성의 바다 OS는 C++ 언어로 개발된다[3]. 본 논문에서는 이러한 개념을 모바일 도메인에 적용하여 다른 아키텍처위의 운영체제에서는 동작하지 않는 시스템을 설계 단계부터 구현까지 다시 반복하여 개발 단계를 거치는 문제점을 해결하고자 접근 방법으로 채택한다.

2.3 안드로이드 모바일 오픈소스 아키텍처

Anvaari(2010)에 의해 모바일 소프트웨어 플랫폼을 위한 아키텍처 오픈니스 모델이 소개되었다.

모바일 소프트웨어 플랫폼은 어플리케이션 개발자들이 모바일 소프트웨어의 기능을 어떻게 확장하

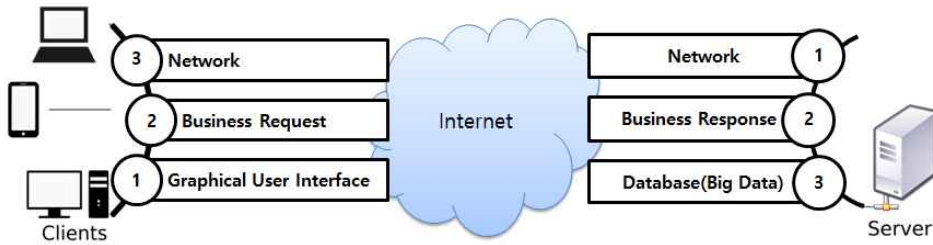


Fig. 1. Basic Structure of the Suggested Framework.

고 플랫폼 컴포넌트들 간의 인터페이스를 정의한다. 정의된 레이어의 모델은 다음과 같이 정의된다. 첫째, 어플리케이션 레이어는 두 개의 서브레이어로 나뉜다. 네이티브 어플리케이션과 확장된 어플리케이션으로 나뉜다. 네이티브 어플리케이션은 기존의 생산자에 의해 개발된 어플리케이션을 말한다. 확장된 어플리케이션은 어플리케이션 개발자와 장비 사용자들에 의해 개발된 어플리케이션을 말한다. 둘째, 미들웨어 레이어는 핵심 라이브러리와 데이터 저장소와 가상머신과 같은 플랫폼의 서비스를 구성한다. 마지막으로 커널 레이어는 디바이스 드라이브, 전원 관리 프레임워크, 보안 프레임워크처럼 플랫폼의 하위 레블을 구성하는 플랫폼의 핵심이다. 이러한 구조는 모바일 플랫폼들의 오픈 소스를 사용하는데 있어서 개념을 명료화하고 변형 가능한 컴포넌트를 지원하는 방법을 찾기 위한 노력이라 할 수 있다[4].

3. MDA를 이용한 크로스 프레임워크

3.1 제안 프레임워크 개요

웹서비스를 바탕으로 기존 컴퓨터에서 이용되던 모든 서비스는 모바일 디바이스에서 이용할 수 있는 서비스로 변환되고 있다. Fig. 1은 본 논문에서 제안하는 프레임워크의 기본구조를 나타내고 있다. 모바일 소프트웨어 구조는 사용자 측면과 서버 측면으로 크게 나눌 수 있다. 특히 그래픽 유저 인터페이스는 사용자의 요구사항을 받아들이고 사용자가 요구하는 비즈니스 앱을 실행하여 클라이언트와 서버사이의 동기화를 네트워크 환경(인터넷 환경)에서 구현한다.

Fig. 2는 본 논문에서 제안하는 개념을 간략히 보여준다. 특정 기능에 대한 요구사항을 분석하고 그 기능에 대한 컴포넌트를 추출하여 UML을 이용한 CIM(Computation Independent Model)단계와 PIM

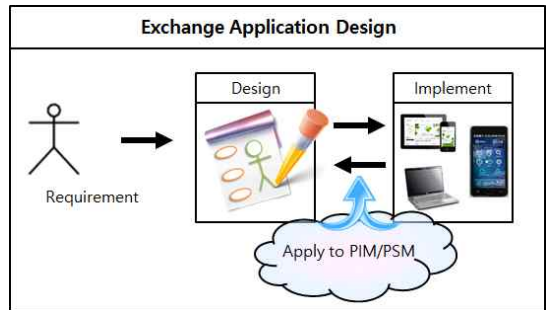


Fig. 2. MMCF(MDA Mobile Cross Framework) Design.

(Platform Independent Model)단계의 결과물을 산출한다. CIM은 컴포넌트 기술 요소의 표준 메타모델을 정의하여 호환성과 재사용성을 보장하기 위해 적용한다. 본 논문의 적용사례로 비중속적인 플랫폼에 초점을 두어 비즈니스 요구사항의 기능적 설계를 CIM단계에서 표현한다.

3.2 제안 프레임워크 적용 프로세스

현재 사용되고 있는 모바일 어플리케이션은 다양한 운영체제에 따라 Tizen의 C++, iOS의 Objective-C, Android의 Java와 같이 다양한 언어를 사용해야 하는 네이티브 어플리케이션과 운영체제를 구분하지 않고 웹브라우저에서 범용적으로 표현할 수 있는 HTML5를 이용한 모바일 웹 어플리케이션으로 나눌 수 있다. 하지만 클라이언트와 서버 두 가지 측면의 모바일 어플리케이션은 모두 자바언어에 근거를 두고 개발된다[5-6].

사용자의 요구사항에 따라 설계되고 특정 언어를 사용하여 개발되는 어플리케이션이 있다고 가정하자. 이 어플리케이션은 모바일 앱 개발 프로세스에 따라 개발 초기 단계인 설계부터 구현단계과정을 거쳐 완성된다. 하지만 개발자가 동일한 기능을 수행하는 어플리케이션을 안드로이드 운영체제와 아이

폰 운영체제처럼 서로 다른 플랫폼에 적용하도록 설계 및 구현한다면 모바일 앱 개발 프로세스를 동일한 방법으로 설계단계부터 서로 다른 두 가지로 개발해야하는 문제점이 있다. 본 논문에서 제시하는 프레임워크를 적용하면 이러한 문제점을 해결할 수 있다. Fig. 3에서는 MDA기반 크로스 프레임워크에 대한 설계 과정을 나타낸다. 사용자가 필요로 하는 기능별로 컴포넌트를 단계별로 분류한 후 명세화 작업을 한다. 명세화 작업은 MDA 접근 방법에 대한 메타모델을 이용하여 나타낸다[7-8]. 이러한 요구사항의 설계과정을 UML(Unified Modeling Language)을 이용하여 나타낸다. 여러 요구사항을 명세화한 메타모델 자료는 구현단계에서 사용자가 플랫폼에 맞게 플랫폼을 구분하여, 그 기능이 구현되도록 해당하는 개발 언어로 변환하면 다양한 운영체제에 맞게 동일한 서비스의 설계와 결과물을 재사용할 수 있는 장점이 있다.

4. 제안 프레임워크 사례연구

본 논문에서는 모바일 통신 과정 중 발생할 수 있

는 수신부재에 대한 처리 방법을 명세화한다.

Fig. 4에서는 구조적 개발 방법 중 N-tire 기반의 개발 구조를 적용하여 사용자 인터페이스 계층과 비즈니스 계층 그리고 구현계층으로 나누어 프레임워크의 계층도를 나타낸다. 특히, 비즈니스 계층은 PIM모델과 PSM모델로 이 영역을 크로스 개발 영역으로 구분한다.

Fig. 5에서는 소프트웨어 개발 방법론의 기본 개발 과정에 MDA 개발 프로세스를 접목하여 크로스 프레임워크 설계 과정을 나타낸다.

4.1 CIM(Computation Independent Model) 단계

개발하고자 하는 시스템의 도메인과 사용자 그리고 기반환경을 고려하여 요구사항을 분석하고 시나리오를 작성함으로써 개발 목표를 명확히 정의한다. 사용자관점에서 시스템이 어떻게 동작하는지를 결정하여 전체 시스템 활동 전반을 스토리텔링 기법으로 요구사항 시나리오를 아래와 같이 작성한다. 모바일 통신기능을 이용하는 사용자는 송신자와 수신자로 구분한다. 송신자는 모바일 기기를 이용하여

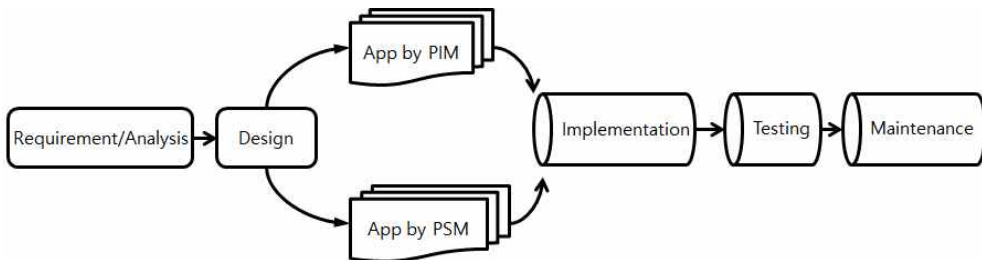


Fig. 3. MMCF Design Process.

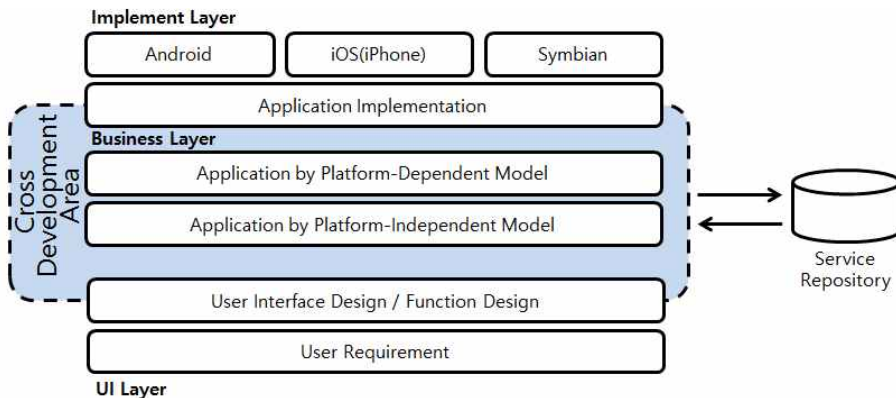


Fig. 4. Architecture of a Realization of MDA Mobile Cross Framework.

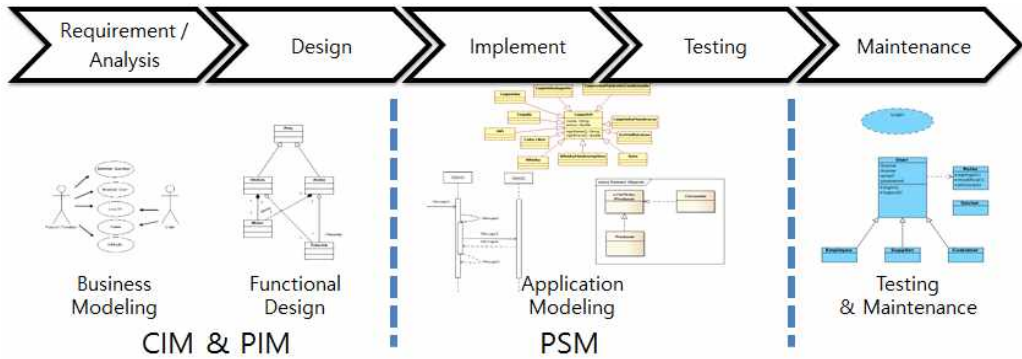


Fig. 5. MDA Mobile Cross Framework Design Process.

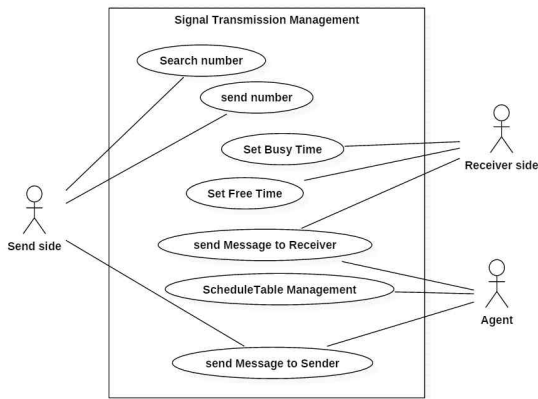


Fig. 6. Communication Management.

연락처를 검색하고 선택한 연락처로 송신을 시도한다. 정상적인 통신의 경우, 수신자가 수신에 용이한 경우는 현재 우리가 사용하는 통신시스템과 차이가 없다. 하지만 수신자가 수신에 불가능한 상태일 경우 상태 감시 에이전트가 수신자의 통신 가능한 스케줄을 검색하여 송신자에게 수신에 가능한 시간대와 현재 수신에 불가능한 이유를 알려주는 메시지를

발송하며 수신자에게는 송신자의 발신시간을 메시지로 남기게 된다. 이러한 시나리오를 Fig. 6에서 유즈케이스 다이어그램으로 나타낸다. Fig. 7은 송신자와 수신자와의 관계를 클래스 다이어그램으로 나타낸다[9-10].

4.2 PIM(Platform Independent Model)단계

송신자와 수신자 그리고 상태감시 에이전트가 상호작용하는 메시지의 흐름을 모델링한 시퀀스 다이어그램을 Fig. 8에서 나타낸다. 요구분석 단계에서 제시한 시나리오를 바탕으로 시퀀스 다이어그램을 나타낼 수 있다[5]. 수신자는 자신의 통신 부제시간을 스케줄 테이블에 등록하고 송신자가 통신을 시도할 경우 상태감시 에이전트는 수신자가 등록해놓은 스케줄테이블을 검색하여 수신자의 상태를 검사한다. 이때 수신자가 응답이 불가능한 상태라면 수신에 불가능한 이유와 수신에 가능한 시간을 문자메시지로 송신자에게 문자메시지로 전송한다. 동시에 수신자에게는 송신자의 정보를 메시지로 보낸다. Fig. 9는 사건과 상태를 관련지은 통신 흐름 형태를 상태

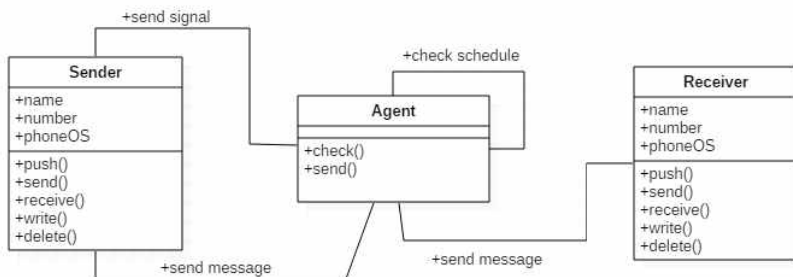


Fig. 7. Relationship Model.

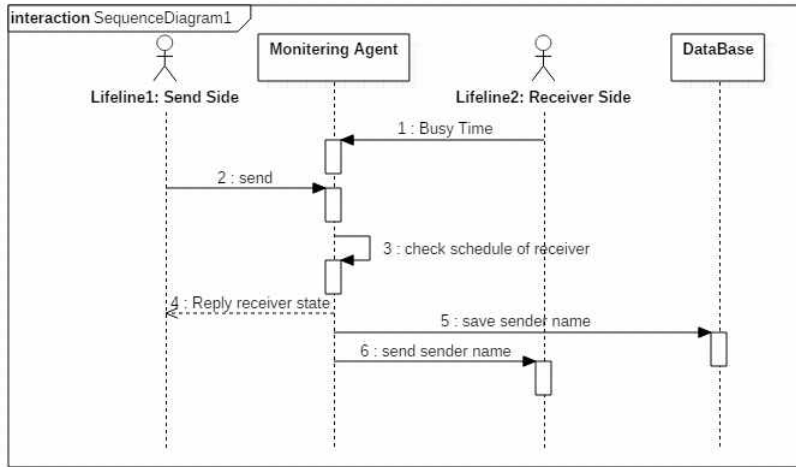


Fig. 8. Message Flow Modeling.

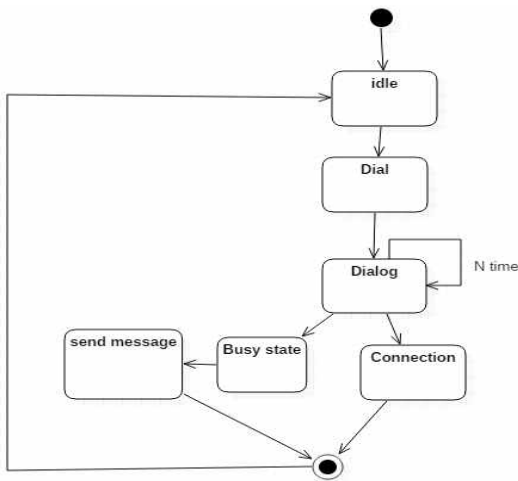


Fig. 9. Communication Flow State.

다이어그램으로 나타낸다.

4.3 기대효과

소프트웨어 품질 특성은 기능적 요구사항에 대한 사용자의 호응도를 중요하게 생각하지만 일반적으로 비기능적인 요구사항에 대한 품질 평가 요소들을 많이 언급한다[4]. 본 논문에서 제시한 MDA기반 크로스 프레임워크의 기대효과는 여러 항목별로 살펴볼 수 있으나 품질평가 요소 항목에 기반을 두어 앱 개발의 중요한 사항들을 나열하였다. Table 1에서는 본 논문에서 제안하는 MMCF는 구조적 개발 방법론을 비교하였다. 구조적 개발 방법은 하향식

Table 1. The performance for proposed MCF Based Development Method

Category	Structured Development Method	MCF Based Development Method
Availability	○	○
Modifiability	×	○
Usability	×	○
Cost	×	○
Integration with Big Data	×	○
United Conceptual	×	○

(top-down)방법으로 시스템을 기능 관점에서 정보의 흐름을 나타낸다. 또한 컴포넌트 단위로 내부 설계구조와 이들의 상호연결 구조에 대한 개발방법이다. 모바일 어플리케이션의 개발자 관점에서 구조적 개발 방법과 MDA기반 앱 개발 크로스 프레임워크를 유용성, 수정가능성, 편리성, 비용, DB통합, 시스템 통합성 측면으로 비교했을 때, 시스템의 구성 요소들과 구조적인 관계 그리고 시스템 개발 특성 면에서 재사용성과 편리성, 비용절감에서 기대효과를 높일 수 있다.

5. 결 론

최근 모바일 어플리케이션 개발방법은 운영체제에 따라 다양한 언어로 개발되어 기존의 애플스토어에

저장된 후 앱 사용자는 자신이 보유한 하드웨어 기종에 맞는 앱을 선택하여 사용한다. 이는 동일한 기능을 수행하는 어플리케이션 일지라도 다양한 기종과 서로 다른 플랫폼에 따라 각각 다르게 개발되어야 한다는 문제점이 있어, 소프트웨어공학에서 추구하는 재사용성과 유지보수성 그리고 이식성을 저하시키는 요인이 된다. 이와 같은 문제점을 해결하려면, 어플리케이션 개발 시 개발 초기 설계단계부터 플랫폼에 비종속적인 설계모델을 적용하고, 동일한 기능의 모바일 앱 설계 결과물이 저장소에 저장된 후 플랫폼에 종속적인 설계모델로 변환하여 재사용할 수 있다면 개발비용과 유지보수 비용을 절감할 수 있을 것이다. 이를 위해서는 특정 시스템 환경에 종속되지 않는 독립적인 기능 명세방법이 필요하고, 그 앱을 개발하는 표준화된 모바일 소프트웨어 개발 프로세스가 필요하다. 본 논문에서는 서로 다른 플랫폼을 사용해야 하는 모바일 앱의 특성을 고려하여 MDA기반 개발 프로세스를 적용하여 특정 시스템 환경에 종속되지 않는 독립적인 기능을 명세화하고, 그 앱을 개발하는 표준화된 모바일 소프트웨어 개발 프로세스를 제안하고 'MDA기반 모바일 크로스 프레임워크'로 명명하였다. 개발될 모바일 앱은 개발 초기 설계단계에서 플랫폼에 비종속적인 설계모델을 적용하고, 이 설계모델을 저장소에 저장시키면 앱 구현 단계에서 그 메타모델을 이용하여 플랫폼에 종속적인 설계모델로 변환 후 구현할 수 있을 것이다. 이 방법을 적용하면 개발비용과 유지보수 비용 면에서 절감할 수 있다고 판단된다. 향후연구로 모바일 어플리케이션 설계과정인 PIM단계에서 표현되어야 할 표준 저장소와 모델 구조정의 메타모델을 보다 자세히 표현하고, PSM단계에서 실제 변환과정이 이루어지는 구현단계에 대하여 연구하고자 한다.

REFERENCE

- [1] E. Cho, "Design of a Metamodel for the Development Process of a Mobile Application," *Journal of the Korea Academia-Industrial Cooperation Society*, Vol. 15, No. 8, pp. 5248-5255, 2014.
- [2] W.S. Kim, O.C. Kwon, and K.S. Shin, "An Introduction to the Model Driven Architecture," *Journal of European Telecommunications Standards Institute*, Vol. 17, No. 6, pp. 11-19, 2002.
- [3] S. Han, Y. Son, and Y. Lee, "Design and Implementation of the Smart Virtual Machine for Smart Cross Platform," *Journal of Korea Multimedia Society*, Vol. 16, No. 2, pp. 190-197, 2013.
- [4] S.T. Abadi, *Towards a Generic Reference Architecture for Mobile Applications*, Bachelor's Thesis of University of Gothenburg, Chalmers University of Technology, 2011.
- [5] W. Jung, J. Oh, and D. Yoon, "Design and Implementation of Hybrid Mobile App Framework," *Journal of Korea Institute of Information and Communication Engineering*, Vol.16, No. 9, pp. 1990-1996, 2012.
- [6] C. Bouras, A. Papazois, and N. Stasinou, "A Framework for Cross-platform Mobile Web Applications Using HTML5," *Proceeding of International Conference on Future Internet of Things and Cloud*, pp. 420-424, 2014.
- [7] B. Gallina, M.A. Javed, F.U. Muram, and S. Punnekkat, "A Model-driven Dependability Analysis Method for Component-based Architectures," *Proceeding of Euromicro Conference on Software Engineering and Advanced Applications*, pp. 233-240, 2012.
- [8] E. Breton and J. Bezivin, "Model-Driven Process Engineering," *Proceeding of Computer Software and Applications Conference*, pp. 225-230, 2001.
- [9] P. Braun and R. Eckhaus, "Experiences on Model-driven Software Development for Mobile Applications," *Proceeding of Engineering of Computer Based Systems, Annual IEEE International Conference and Workshop on the*, pp. 490-493, 2008.
- [10] S. Deelstra, M. Sinnema, J.V. Gorp, and J. Bosch, "Model Driven Architecture as Approach to Manage Variability in Software Product Families," *Proceeding of Workshop on Model Driven Architecture: Foundations and Applications*, pp. 109-114, 2003.



송 유 진

2003년 대구가톨릭대학교 컴퓨터
정보통신공학부 (학사)
2006년 대구가톨릭대학교 전자계
산교육전공 (석사)
2008년 3월~현재 경북대학교 IT
대학 컴퓨터학부 (박사수
료)

관심분야: Component Based Design, Model Driven
Architecture, Mobile Application Logic
Analysis, software testing



한 덕 수

1988년 금오공과대학교 전자계산
기공학과 (학사)
1998년 대구가톨릭대학교 전산통
계학과 (석사)
2003년 연세대학교 컴퓨터과학과
(박사)

1992년 6월~현재 육군3사관학교 컴퓨터공학과 교수
관심분야: Multimedia, Virtual Reality, HCI, e-Learning,
Software Engineering



이 은 주

1997년 서울대학교 계산통계학과
(학사)
1999년 서울대학교 전산학과
(석사)
2005년 서울대학교 전기컴퓨터공
학부 (박사)

2005년 3월~2005년 10월 서울대학교 공과대학 BK 박사
후 연구원

2005년 11월~2006년 2월 삼성종합기술원 전문연구원
2006년 3월~현재 경북대학교 IT대학 컴퓨터학부 (전임
강사, 조교수, 현재 부교수)

관심분야: Mining software repository, Change pattern,
software metric, software evolution