

# 정보과학회논문지 : 컴퓨팅의 실제 및 레터

Journal of KIISE : Computing Practices and Letters

VOLUME 20, NUMBER 4, APRIL 2014

## 멀티미디어 통신

- 음성통신을 위한 TDMA 기저대역 중계모뎀 설계 ..... 배용욱, 김중건, 안병철 207

## 정보분석 서비스

- 영화 평점에서 악의적 평점제공자 판별 및 실플평점 유추 ..... 김경민, 안무혁, 이윤호 213

## 레터 논문

### 국방 소프트웨어

- 해군 함정 컴퓨팅 체계 고가용성과 최적화를 위한 클라우드 컴퓨팅 적용 모델 ..... 엄준호, 오상윤 219

### 영상처리

- 색상 패턴을 고려한 예술적 텍스처 전이 기법 ..... 박지성, 강동완, 윤경현 224  
유화 스타일화 벡터 이미지 생성기법 ..... 윤종철, 이인권 229

### 소프트웨어 공학

- 셀렉션 코딩 패턴의 소모 전력 효율 향상을 위한 코딩 가이드라인 ..... 박재진, 김두환, 홍장의 233  
이종 스마트폰 앱 개발환경 구성을 위한 모델변환 적용 방법 ..... 손현승, 김우열, 김영철 238

### 웹 서비스

- 모바일 환경에서 시간 속성을 고려한 효율적인 위치 기반 소셜 검색 ..... 복경수, 안민제, 임종태, 유재수 243

### 인공지능

- 방향 및 위상 관계들 간의 교차-일관성 검사를 지원하는 정성적 공간 추론기 ..... 남상하, 한상두, 차영일, 김인철 248

### 인터넷 소프트웨어

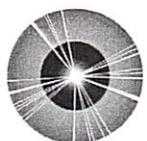
- 도로 접근성 표현을 위한 왜곡 지도 시각화 기술 ..... 조광수, 홍성수, 윤종철 253

### 정보 연계 모델

- 뇌의 기억 인출에 대한 유효 EEG 연결성 분석 ..... 이충연, 장병탁 257

### 클라우드 컴퓨팅

- 클라우드 환경에서 고해상도 이미지 프라이버시 보호를 위한 분산 암호화 기법 ..... 정장영, 홍영식 262



한국정보과학회

KOREAN INSTITUTE OF INFORMATION SCIENTISTS AND ENGINEERS

# 이종 스마트폰 앱 개발환경 구성을 위한 모델변환 적용 방법

(A Method of Applying Model Transformation for Development Environment Configuration of Heterogeneous Smartphone App)

손 현 승 <sup>†</sup> 김 우 열 <sup>††</sup>

(Hyun Seung Son) (Woo Yeol Kim)

김 영 철 <sup>†††</sup>  
(R. Young Chul Kim)

**요약** 다양한 스마트폰 플랫폼을 가진 안드로이드, 아이폰, 윈도우즈 폰과 같이 SW의 플랫폼은 매우 이질적인 특성을 갖기 때문에 개발되는 회사나 사람에 따라, 똑같은 콘텐츠의 어플리케이션을 재사용하지 못하고 다른 플랫폼 개발시 새로운 코드를 작성해야만 한다. 이 문제 해결을 위해 최근에는 브라우저 기반의 하이브리드 앱이 주목받고 있다. 하지만 하이브리드는 간단한 앱에도 브라우저를 로드해야 해 오버헤드가 크고 네이티브 앱의 UI와 같게 만들기 위해 많은 노력이 소모된다. 이 논문에서는 기존 연구를 토대로 이종 스마트폰 앱 개발환경 구성을 위한 모델변환의 적용 방법을 제시한다. 특히 이종 스마트폰 앱을 개발에 프

로젝트, UI, 코드의 3가지 요소가 필요하지만, 제안한 방법은 개발환경 구성에 필요한 프로젝트 파일에 대해서만 다룬다. 제안한 방법은 각 플랫폼에 프로젝트 파일들을 분석하여 공통과 가변 요소를 추출하는 분석 단계와 이를 이용하여 메타모델링과 변환규칙을 만드는 설계 단계로 구성되어 있다. 적용사례에서는 이 방법이 이종 플랫폼의 프로젝트 파일을 동시에 생성 가능한지를 보인다.

**키워드:** 모델변환, 메타모델링, 스마트폰, 다중플랫폼, 재사용

**Abstract** The software platforms like Android, iPhone, and Windows Phone with the various smartphone platforms have very heterogeneous characteristics in each platform, which can not reuse software dependant on the company or person, and must be rewrite new code for other software platforms without reusability even though the same content applications. In recent years to solve this problem, browser-based hybrid apps is interested. But the hybrid has many overhead on even simple software app to load the browser, and consumes a lot of effort to make UI like a native app. In this paper, we propose a method applying model transformation for development environment configuration of heterogeneous smartphone app based on our previous research. To develop heterogeneous smartphone app is required with three element as project, UI, and code. But this paper is only limited to project files required to configure the development environment in our proposed method. This method consists of 1)the analysis step to extract the common and variable elements through analyzing project files for each platform, and 2)the design step to create metamodel and transformation rules from the analyzed results. With a case study, we show how the proposed method can create project files at the same time.

**Keywords:** model transformation, metamodeling, smartphone, multi-platform, reuse

## 1. 서론

전 세계적으로 스마트폰의 관심과 판매 증가로 각 스마트폰 제조 및 공급 회사마다 플랫폼을 선점하기 위해서 플랫폼 전쟁이 시작되었다[1]. 2012년 기준 안드로이드 약 55%, 아이폰 약 23% 플랫폼이 점유율을 보이고 있고 MS의 윈도우즈 폰, 삼성의 태이젠 등 새로운 플랫폼들도 추가되고 있다[2].

플랫폼의 종속적인 특징 때문에 특정 플랫폼기반의 소프트웨어는 다른 플랫폼으로 재사용하지 못한다. 이 문제의 해결을 위해 브라우저 기반의 하이브리드 앱 개발 방법이 등장하였다. 하지만 하이브리드 방법은 간단한 앱에도 브라우저를 실행해야해 오버헤드가 크고 플랫폼 별로 구현된 기능들이 제한적이다. 또한 네이티브 앱의 UI와 같게 만들기 위해 초기에 많은 비용이 소모

† 본 연구는 미래창조과학부 및 한국산업기술평가관리원의 산업원천기술개발사업(10035708, 고신뢰 자율제어 SW를 위한 CPS(Cyber-Physical Systems) 핵심 기술 개발)과 2013년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업입니다(2013R1A1A2011601).

†† 이 논문은 제40회 추계학술발표회에서 '이종 스마트폰 앱을 위한 모델변환 도구 개발'의 제목으로 발표된 논문을 확장한 것입니다.

††† 학생회원 : 홍익대학교 컴퓨터정보통신공학과 SE연구실  
hson@live.co.kr

††† 비회원 : 대구교육대학교 컴퓨터교육과 교수  
john@dnue.ac.kr

††† 정회원 : 홍익대학교 컴퓨터정보통신공학과 SE연구실 교수  
bob@hongik.ac.kr  
(Corresponding author임)

논문접수 : 2014년 1월 13일  
심사완료 : 2014년 2월 5일

Copyright©2014 한국정보과학회 : 개인 목적이나 교육 목적인 경우. 이 저작물의 전체 또는 일부에 대한 복사본 혹은 디지털 사본의 제작을 허가합니다. 이 때, 사본은 상업적 수단으로 사용할 수 없으며 첫 페이지에 본 문구와 출처를 반드시 명시해야 합니다. 이 외의 목적으로 복제, 배포, 출판, 전송 등 모든 유형의 사용행위를 하는 경우에 대하여는 사전에 허가를 얻고 비용을 지불해야 합니다.

정보과학회논문지 : 컴퓨팅의 실제 및 레터 제20권 제4호(2014.4)

되는 문제 등 아직은 부족하다[3].

MDD(Model Driven Development)[4]는 모델을 재사용과 코드 자동생성으로 생산성을 높일 수 있는 방법이다. 우리는 이러한 장점 때문에 MDD를 임베디드에 적용한 e(Embedded)-MDD를 제안하였다[5,6]. e-MDD는 다중의 영역으로 모델로 분리 가능해 플랫폼 독립모델을 더 많이 재사용할 수 있고 자동화로 이종의 플랫폼의 소프트웨어를 빠르게 개발할 수 있다.

스마트폰 플랫폼들은 운영체제, 개발언어, UI 코드, 개발환경 등 서로 매우 다른 구성을 가지고 있다. 예를 들어 안드로이드는 자바, 아이폰은 오브젝티브-C, 윈도우즈 폰은 C# 등 개발 코드가 다르고 UI 언어는 XML로 같지만 구조는 다르다. 이러한 특성 때문에 그림 1과 같이 모델변환은 모델에서 코드생성 뿐만이 아니라 이와 관련되어 있는 프로젝트, UI 파일들을 모두 생성해야 한다. 여기서 UML은 코드 생성을 위한 것이고 UI는 각 플랫폼 UI 정보들의 특징을 추출한 추상화한 UI 모델로 이종의 UI 모델로 변환하기 위한 것이다. 완전한 변환을 위해서는 각 요소마다의 모델변환 방법이 필요하지만, 여기에서는 모든 내용을 다루기는 어려워 이 요소들 중 개발환경 구성에 필요한 프로젝트 파일(그림 1의 음영부분)의 모델변환에 대해서만 다룬다.

이 논문에서는 UI, 코드, 프로젝트 중 이종 스마트폰 앱 개발환경 구성을 위해 필요한 프로젝트 파일에 대한 모델변환의 적용 방법을 제안한다. 제안한 방법은 각 플랫폼에 프로젝트 파일들을 분석하여 공통과 가변 요소를 추출하는 분석 단계와 이를 이용하여 메타모델링과 변환규칙을 만드는 설계 단계로 구성되어 있다. 그리고 이렇게 만들어진 메타모델과 모델변환규칙을 모델변환 엔진으로 모델변환을 수행하면 이종의 프로젝트 파일들을 생성할 수 있다. 적용사례에서는 이 방법이 이종 플랫폼(안드로이드, 아이폰, 윈도우즈 폰)의 프로젝트 파일을 동시에 생성 가능한지를 보인다.

이 논문의 구성은 다음과 같다. 2장에서는 관련연구로 기존의 크로스 플랫폼 개발 도구들을 비교를 한다. 3장

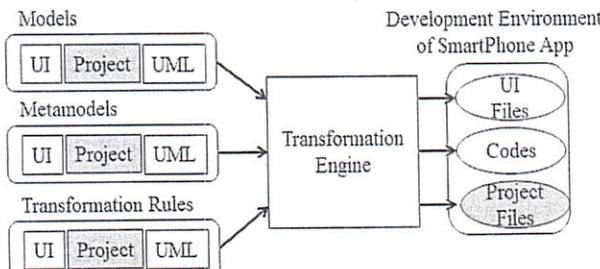


그림 1 이종 스마트폰 앱을 위한 모델변환의 구조  
Fig. 1 The structure of model transformation for heterogeneous smartphone app

에서는 제안한 모델변환 방법에 대해서 설명하고, 4장에서는 적용사례를 보여준다. 마지막으로 5장에서는 결론 및 향후 연구에 대해서 언급한다.

## 2. 관련 연구

국내외 크로스 플랫폼 개발 도구는 스마트폰의 발전에 따라 점점 더 다양해지고 있다. 크로스 플랫폼 개발의 대표적인 개발 도구에는 PhoneGap[7], m-Bizmaker[8], Appspresso[9]가 있다. PhoneGap은 웹을 이용하여 Negative 앱으로 변형시켜주는 도구로써 아이폰 뿐만 아니라 다양한 플랫폼을 지원하는 것이 특징이다. m-Bizmaker는 비즈니스 어플리케이션 제작에 있어 효율성을 낼 수 있도록 편리한 기능과 사용자 중심의 UI를 제공하고 국내 기업에서 제공하는 것이 특징이다. Appspresso는 웹 기술로 개발한 소스를 각 모바일 플랫폼 별 런타임으로 감싸 네이티브 앱으로 빌드하는 하이브리드 방식의 크로스 플랫폼 모바일 프레임워크이다.

기존의 크로스 플랫폼 지원을 위한 하이브리드 방식은 웹(Web)을 앱(App)처럼 사용할 뿐, 네이티브 앱과는 많은 차이가 있다. 이러한 웹기반 앱들은 중간에서 통신 담당을 수행해주는 미들웨어에 많은 의존을 하게 되고 웹의 기능의 한계를 가지고 있으며 자바스크립트를 통해서 모든 처리를 해야 한다. 또한 하이브리드 기반 개발은 HTML을 앱처럼 보이게 하기위해서 UI 작업에 상당한 노력이 소모되고 간단한 앱도 브라우저를 실행해야 하기 때문에 높은 사양의 디바이스가 요구된다.

## 3. 개발환경의 모델변환 방법

### 3.1 모델변환 적용 방법

모델변환은 1)모델에서 모델과 2)모델에서 텍스트의 두 가지 방법 있는데, 모델에서 텍스트로의 변환은 출력되는 값이 텍스트로 메타모델이 필요 없다.

프로젝트 파일들은 그 구조가 복잡하고 간단한 변환을 위해서 다양한 파일들의 메타모델을 설계하기 어렵기 때문에, 이 논문에서는 모델에서 텍스트 변환을 사용한다. 모델에서 텍스트 변환방법을 사용하기 위해서는 메타모델과 변형규칙의 두 가지 요소가 필요하다. 그러므로 제안한 모델변환 적용 방법은 그림 2와 같이 각 플랫폼에 프로젝트 파일들을 분석하여 공통과 가변 요소를 추출하는 분석 단계와 이를 이용하여 메타모델링과 변환규칙을 만드는 설계 단계로 구성한다. 공통과 가변 요소를 추출하는 분석 단계에서는 각 플랫폼의 프로젝트 파일들을 비교하여 같은 정보와 다른 정보를 분리하고 의미 있는 요소들을 추출해낸다. 메타모델링과 변환규칙 설계 단계에서는 전 단계의 분석 데이터를 이용하여 메타모델링과 변환규칙을 만든다.

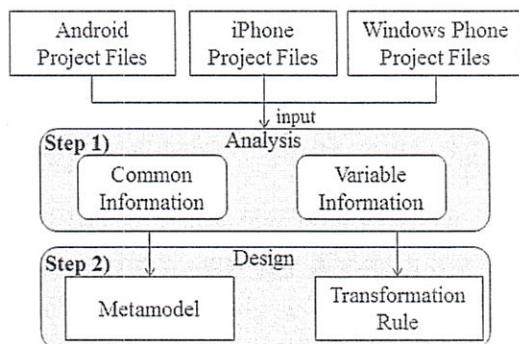


그림 2 프로젝트 파일에 모델변환 적용 방법

Fig. 2 The method applying model transformation to project files

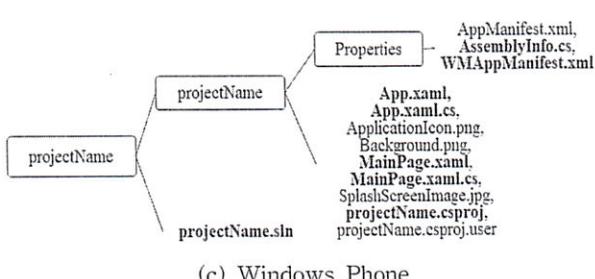
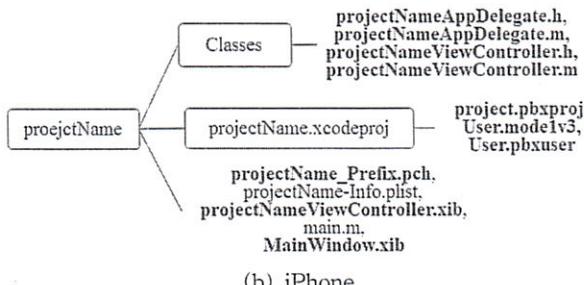
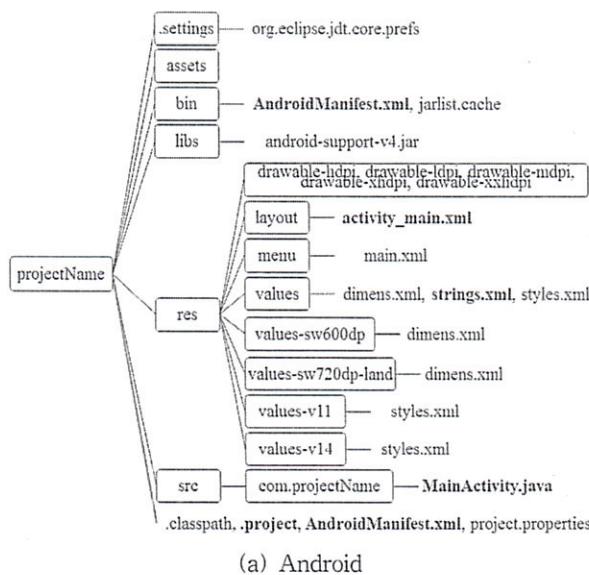


그림 3 각 플랫폼의 파일구조

Fig. 3 The file structure of each platform

### 3.2 공통과 가변 요소를 추출하는 분석 단계

공통과 가변 요소를 추출하기 위해서 우리는 각 플랫폼의 개발도구에서 생성하는 초기 프로젝트 파일들을 수작업으로 비교 분석하였다. 그 결과 그림 3과 같이 파일의 구조를 얻을 수 있었고 변경되는 데이터는 블드체로 표현하였다. 각 플랫폼의 분석한 결과 3가지의 공통적인 특징들을 찾아낼 수 있었다. 첫 번째로는 각 프로젝트의 이름이 파일 내에 다수로 입력된다는 것과 두 번째로는 파일과 디렉토리 생성에도 프로젝트이름이 사용되는 것, 그리고 마지막으로 변경되지 않는 파일들이 있다는 것이다.

각 플랫폼의 가변 요소는 플랫폼마다 생성되는 파일의 내용과 디렉토리 구조가 다른 것이다. 공통 요소로 추출된 것은 메타모델로 설계가 가능하고 가변 요소는 모델변환규칙을 이용하여 변환을 수행하면 각 플랫폼의 구조의 데이터를 생성할 수 있다.

### 3.3 메타모델과 모델변환규칙 설계 단계

메타모델은 입력되는 모델의 스키마로 데이터의 구조를 결정하게 된다. 앞에서 분석한 공통요소를 기반으로 메타모델을 설계하면 그림 4와 같다. 이 메타모델은 그림 3의 파일 구조를 메타모델링한 것이다.

Project는 생성될 프로젝트의 이름을 포함하는 루트 노드이다. 프로젝트 내부에는 여러 개의 디렉토리를 포함할 수 있고 하부에는 여러 개의 파일과 디렉토리를 포함할 수 있다. 서브노드의 깊이는 입력되는 데이터에 따라 달라진다. isChangeName은 이름이 변경될 경우를 나타내는 속성 값으로, 이 값이 참(True)이 되면 changeNameInfo에 변형될 이름의 형식정보를 입력한다. 형식 정보는 문자열로 입력된다.

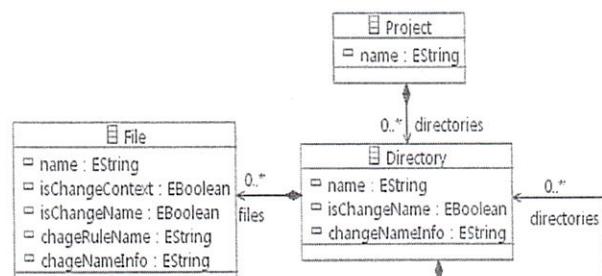


그림 4 프로젝트의 메타모델

Fig. 4 The project of metamodel

이 문자열은 모델변환시에 엔진에 의해서 입력된 값의 “[]”사이의 값은 속성 값으로 대체되고 나머지 값은 그대로 출력된다. 예를 들어 ProjectName이 “project”이고 Name이 “name”이라면, 아래와 같이 입력데이터 1이 2로 변형된다.

1	[ProjectName]_[Name]
2	project_name

File에 isChangeContext는 데이터가 변형될지의 유무를 설정하는 값이다. isChnageContext의 값이 참일 때, changeRuleName에 데이터를 변형시킬 규칙의 이름이 입력되고 이 이름은 모델변환엔진과 연동되어 변형을 위한 기본정보를 제공한다.

이렇게 설계된 메타모델은 EMF(Eclipse Modeling Framework)[10]를 통해서 이를립스 플러그인으로 생성 가능하다. 이 플러그인을 이용하면 모델에 데이터를 입력할 수 있다. 그림 5는 그림 3(c)의 윈도우즈 폰의 파일 구조를 메타모델 기반으로 데이터를 입력한 것이다.

모델에서 텍스트 변환규칙은 모델-텍스트 변환 언어[11]를 이용해서 작성한다. 각 플랫폼(안드로이드, 아이폰, 윈도우즈 폰)의 변형 규칙을 정리하면 표 1과 같다. 변형규칙은 프로젝트 이름 입력, 코드 생성, UI 생성의 3가지 형태로 구분된다. 코드와 UI의 생성을 다루지 않기 때문에 이 부분은 생략한다.

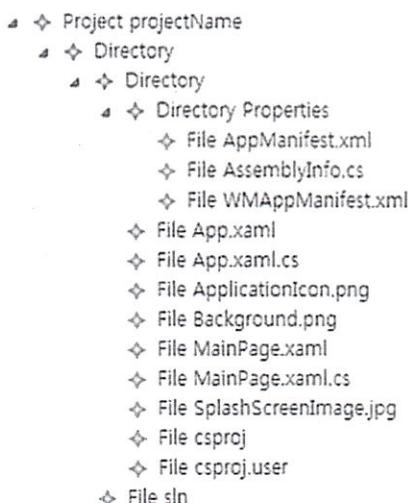


그림 5 윈도우즈 폰 파일 구조의 모델

Fig. 5 The model of Windows Phone's file structure

프로젝트 이름 입력 규칙을 아이폰의 RAppDelegate\_h 규칙으로 예를 들면 그림 6과 같다. 블드로 표현된 [project.name/]은 모델변환시에 프로젝트 이름으로 교체되고 나머지는 그대로 출력된다. 그 외에 다른 규칙들도 모두 이러한 방식으로 기술되어 있다. 모델변환이 수행되는 과정을 종합적으로 설명하면 다음과 같다. 앞에서 정의한 메타모델 기반으로 플랫폼별로 파일구조 모델을 작성한다. 작성한 파일 구조 모델을 모델변환 수행도구에 입력하면 도구는 입력된 모델의 정보를 읽어 깊이 우선 탐색으로 모든 파일 노드를 검사한다.

표 1 모델변환을 위한 규칙  
Table 1 Rules for model Transformation

	Rule Name	Type
Android	RProject	Input project name
	RAndroidManifest	Input project name
	RMainActivity	Generate code
	RActivity_main	Generate UI
	RStrings	Input project name
iPhone	RPrefix	Input project name
	RViewController	Generate UI
	RMainWindow	Input project name
	RAppDelegate_h	Input project name
	RAppDelegate_m	Input project name
	RViewController_h	Generate code
	RViewController_m	Generate code
	RProject_pbxproj	Input project name
	RUser_modelv3	Input project name
	RUser_pbxuser	Input project name
Windows Phone	RSln	Input project name
	RAssemblyInfo	Input project name
	RWMApManifest	Input project name
	RApp	Input project name
	RApp_xaml	Input project name
	R MainPage	Generate UI
	R MainPage_xaml	Generate code
	RCsproj	Input project name

이때 isChangeName의 값이 참이면 changeNameInfo의 정보를 이용하여 이름을 변경하고 isChangeContext의 값이 참이면 changeRuleName의 변형 규칙을 찾아 모델변환을 실행한다. 이러한 방법을 이용하여 서로 다른 이종의 플랫폼의 파일 구조를 유연하게 생성가능하고 프로젝트 환경을 구성할 수 있다.

```

#import <UIKit/UIKit.h>
@class [project.name/]ViewController;
@interface [project.name/]AppDelegate : NSObject
<UIApplicationDelegate> {
    UIWindow *window;
    [project.name/]ViewController *viewController;
}
@property (nonatomic, retain) IBOutlet UIWindow *window;
@property (nonatomic, retain) IBOutlet [project.name/]ViewController *viewController;
@end
  
```

그림 6 아이폰의 모델변환 규칙  
Fig. 6 The model transformation rule of iPhone

#### 4. 적용사례

제안한 방법의 실행을 확인하기 위해서 개발한 HMMT

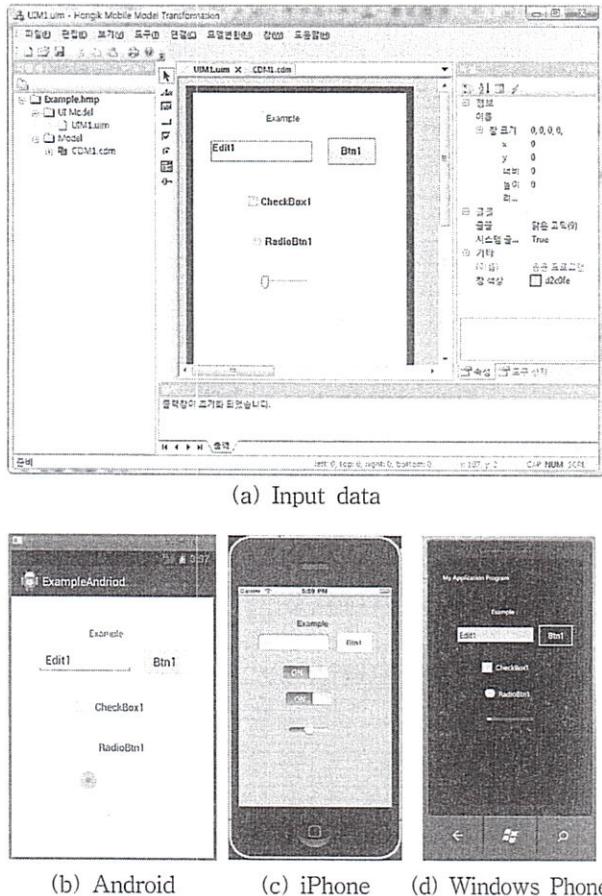


그림 7 모델변환 과정으로 생성된 파일의 각 플랫폼별 실행결과

Fig. 7 The each platform's execution result of created file though model transformation process

(Heterogenous Mobile Model Transformation) 도구를 이용하여 그림 7과 같이 모델 변환을 수행하였다. 각 플랫폼별로 생성하면 UI 모델에서 생성한 모델이 각 플랫폼별로 실행되고 모양이 같은 것을 확인할 수 있고 파일 구조도 앞에서 분석한데로 생성됨을 확인하였다. 안드로이드는 20개의 디렉토리와 18개의 파일, 아이폰은 3개의 디렉토리와 12개의 파일, 윈도우즈 폰은 3개의 디렉토리와 13개의 파일이 각각 생성되었다.

## 5. 결론 및 향후 연구

개발 도구, 프로그래밍 언어, API의 특성이 다르기 때문에 이종의 플랫폼을 동시에 개발하는 것은 매우 어려운 작업이다. 그럼에도 불구하고 이종 플랫폼 개발을 위한 재사용 연구는 끊임없이 이어져 왔었다. 이러한 상호 운영의 문제를 해결하고자 웹 브라우저를 이용한 하이브리드 방식이 개발됐지만 브라우저와 폰을 연결하는 미들웨어에 종속되는 특성, 기존 플랫폼을 최대한 활용하지 못하는 점과 제약점들 때문에 새로운 대안이 필요

하였고 우리는 MDD를 접목하여 스마트폰 플랫폼을 위한 e-MDD 방법을 제안하였다.

스마트폰 플랫폼 분석결과 대부분 UI 편집도구를 기반으로 소프트웨어를 개발하는 구조로 구성되기 때문에, 스마트폰 앱 개발은 코드생성 뿐만이 아니라 이와 관련되어 있는 프로젝트, UI 파일들을 모두 생성해야 한다. 이 논문에서는 이를 실현하기 위한 첫 단계로 UI, 코드, 프로젝트 중 이종 스마트폰 앱 개발환경 구성을 위해 필요한 프로젝트 파일에 대한 모델변환의 적용 방법을 제시하였다. 제시한 방법을 통해서 각 플랫폼에 프로젝트 파일들을 분석하여 공통과 가변 요소를 추출하였고 메타모델링과 변환규칙 생성으로 모델변환을 실행할 수 있었다. 또한 적용사례를 통해 이 방법이 이종 플랫폼(안드로이드, 아이폰, 윈도우즈 폰)의 프로젝트 파일을 동시에 생성 가능한지를 보였다. 제안한 방법을 통해서 이종의 소프트웨어를 동시에 개발하고 재사용 할 수 있어 타 회사 기기간의 콘텐츠의 활용도가 높아질 것으로 기대한다.

향후에는 UI 생성과 코드 생성에 대해서 다루고자 하고 기 개발된 기술로 HMMT 도구를 확장하여 실제 산업 현장에서 적용될 수 있도록 기술을 보완 완성하고자 한다.

## References

- [1] Brad Cho, Platform wars, BOOK21, 2013.
- [2] Yu-Mi Bae, Sung-Jae Jung, Wooyoung Soh, "Trend analysis of Mobile Operating Systems," *Journal of Security Engineering*, vol.9, no.4, pp.343-356, Aug. 2012.
- [3] Junho Jin, Jong-Hong Jeon, Seung-Yun Lee, "Mobile App Development based on HyWAI Framework," *Journal of KIISE : Computer Systems and Theory*, vol.39, no.3, pp.188-195, Jun. 2012.
- [4] B. Selic, "The pragmatics of model-driven development," *Software, IEEE*, vol.20, Issue 5, pp.19-25, 2003.
- [5] Woo Yeol Kim, Hyun Seung Son, Jae Seung Kim, R. Young Chul Kim, "Development of Windows Mobile Applications using Model Transformation Techniques," *Journal of KIISE : Computing Practices and Letters*, vol.16, no.11, pp.1091-1095, Nov. 2010.
- [6] Woo Yeol Kim, Hyun Seung Son, R. Young Chul Kim, "Rule Extraction Method for Model Transformations in Heterogeneous Smartphone Applications," *Information Journal*, vol.16, no.1(B), pp.615-626, Jan. 2013.
- [7] PhoneGap, <http://www.phonegap.com>
- [8] m-Bizmaker, <http://www.mbizmaker.com>
- [9] Appspresso, <http://www.appspresso.com>
- [10] EMF, <http://www.eclipse.org/modeling/emf>
- [11] OMG, "MOF Model to Text Transformation Language," vol.1.0, 2008.