다양한 모바일 플랫폼에 적용 가능한 스마트폰 응용프로그램 개발을 위한 방법 연구

(A Research on Developing Smartphone Applications for Various Mobile Platforms)

한윤선, 홍원기

포항공과대학교 정보전자융합공학부

seon054@postech.ac.kr, jwkhong@postech.ac.kr

요 약

스마트폰을 위한 응용프로그램은 누구나 개발자가 될 수 있으며 자유로운 배포 및 사용이 가능하다. 하지만, 현재 스마트폰 시장은 표준화된 개발 플랫폼과 언어의 부재로 인해 다양한 스마트폰 플랫폼, 개발 언어, 개발 방법이 사용되고 있다. 이러한 상황에서 개발자는 각 플랫폼에 따라 개별적으로 응용프로그램을 개발하여야 하는 문제점이 지적되고 있다. 본 연구에서 정보 모델을 이용한 다양한 모바일 플랫폼에 적용 가능한 스마트폰 응용프로그램 개발 방법을 제시하고 이를 지원하는 도구의 개발 방법을 제안한다. 제안하는 방법은 각 스마트폰 플랫폼에서 추출된 공통점에 기반하여 정보 모델을 설계하고, 이 모델을 바탕으로 각 플랫폼에서 실행 가능한 응용프로그램으로 변환되어 각 플랫폼 환경에서 실행되게 된다.

Keywords: Smartphone, Cross Platform Development, Information Model

1. 서론

기존 일반 핸드폰과 비교하여 스마트폰이 가지는 중요한 차별성 중 하나는 바로 응용프로그램이다. 일반 핸드폰의 경우 응용프로그램은 소수의 회사 혹은 개발자에 의해서만 개발되고 배포되어 왔었다. 이러한 환경으로 인해 사용자들은 다양한 응용프로그램을 사용할 수 없었다. 하지만 스마트폰의 등장으로 인해 이러한 환경이 바뀌었다. 스마트폰 제조사에서 제공하는 편리한 개발환경을 이용하여, 누구나 응용프로그램을 개발하고 배포 할 수 있게 되었다. 스마트 기기들은 누구나 개발자가 될 수 있으며 자유로운 배포 및 사용이 가능하다. 하지만, 현재 스마트폰 시장은 표준화된 개발 플랫폼과 언어의 부재로 인해 다양한 스마트폰 플랫폼, 개발 언어, 개발 방법이 사용되고 있다. 이러한 상황에서 개발자는 각 플랫폼에 따라 개별적으로 응용프로그램을 개발하여야 하는 문제점이 지적되고 있다. 또한 확장을 위해 다른 스마트폰 플랫폼에서 구동 가능한 응용프로그램 출시를 위해서는, 요구사항 명세를 제외한 나머지 모든 부분을 처음부터 새롭게 시작해야 하므로 개발 시간, 인력의 낭비 등을 초래하고 있다. 본질적으로 동일하지만 별도의 스마트폰 플랫폼을 지원하기 위해 추가적으로 개발해야 하는 응용프로그램 제작 시 발생하는 노력과 비용을 줄이는 방법이 필요하다.

스마트폰 응용프로그램 개발에 있어 가장 문제가 되는 것은 스마트폰 플랫폼의 다양성이다. 현재 시장에는 다양한 제조사들이 많은 스마트폰을 출시하고 있다. 이러한 다양한 스마트폰 플랫폼으로 인하여, 개발자는 하나의 응용프로그램 개발 시 다양한 플랫폼에 대응하게 하기 위해서 중복된 노력이 들어가게 된다. 최근 몇몇 외국의 소프트웨어 회사들은 다양한 스마트 플랫폼의 개발을 지원하는 통합 개발 방법 및 환경을 지원해주는 도구들을 발표했다. 이러한 분야에서 가장 핵심적으로 주제는 통합 환경을 통하여 개발된 응용프로그램을 각 스마트폰 플랫폼에서 구동 가능한 형태로 바꿔주는 과정이다. 현재 이러한 변환 과정을 위한 접근법 중 가장 큰 갈래를 이루고 있는 것은 웹 기술 기반의 방법과 스크립트 언어를 이용한 변환 방법이다.

본 연구에서 정보 모델을 이용한 다양한 모바일 플랫폼에 적용 가능한 스마트폰 응용프로그램 개발 방

본 연구는 지식경제부 및 정보통신산업진흥원의 대학 IT 연구센터 지원사업 (NIPA-2012-H0301-12-3002)과 한국연구재단을 통해 교육과학기술부의 세계수준의 연구중심대학육성사업(WCU)으로부터 지원받아 수행되었습니다 (R31-2010-000- 10100-0).

법을 제시하고 이를 지원하는 도구의 개발 방법을 제안한다. 제안하는 방법은 각 스마트폰 플랫폼에서 추출된 공통점에 기반하여 정보 모델을 설계하고, 이 모델을 바탕으로 각 플랫폼에서 실행 가능한 응용프로그램으로 변환되어 각 플랫폼 환경에서 실행되게 된다.

2. 관련연구

단일 개발 결과를 다양한 플랫폼에 적용하기 위한 방법에 대한 연구는 새로운 개념이 아니다. 기존의 일반 컴퓨팅 환경에서도 다양한 플랫폼 환경을 지원하기 위한 크로스 플랫폼 개발(Cross Platform Development)연구가 활발이 있어왔다. 이들 연구의 내용을 살펴 보면, 크게 하드웨어 플랫폼의 다양성, 소프트웨어 플랫폼의 다양성을 극복하기 위한 방법이 주를 이루고 있다. 이러한 연구 결과 중 대표적인 것은 JAVA 프로그래밍 언어로써, JAVA 는 단일 프로그래밍 코드를 Windows, Mac OX, Linux 등과 같은 다양한 운영체제 환경에서 실행 할 수 있도록 지원하고 있다. 일반 컴퓨팅 환경에서 크로스 플랫폼 개발을 지원하기 위해 Cairo, Eclipse, FLTK, fpGUI, GeneXus, GTK+, haXe, Juce 등과 같은 많은 도구들이 현재 사용 가능하다.

최근 기존의 컴퓨팅 환경뿐만 아니라 스마트폰 환경에서도 크로스 플랫폼 개발을 지원하기 위한 연구가 활발히 진행 중에 있다. 스마트폰 환경에서 크로스 플랫폼 지원을 위한 연구 중 큰 갈래를 이루고 있는 두 가지 접근법은 웹 기술 기반의 방법과 스크립트 언어를 이용한 변환 방법이다. 웹 기술에 기반한 방법은 대부분의 핵심 사항들을 HTML5, CSS 와 같은 표준 웹 기반 기술로 구현 한 후, 스마트폰이 기본적으로 제공하는 웹 접근 기능을 이용하여 개발된 기능을 통합한다. 대표적인 방법으로 PhoneGap, Titanium, Appspresso 이 있다. 다른 대표적인 방법으로 자체적으로 개발 언어 및 환경을 구축하고 개발을 위한 API 를 제공하는 방법이 있다. 이러한 방법들은 특정 플랫폼에 종속적이지 않은 스크립트 언어를 사용하여 응용프로그램을 개발하고 그 결과물을 각 플랫폼서 실행 가능한 형태로 변환 시켜주며, 예로 Corona 가 있다.

PhoneGap 은 오픈 소스 프로젝트의 일환으로써 Javascript, HTML5, CSS3 와 같은 웹 기반 기술을 활용하여 크로스 플램폼 개발을 지원하고 있다. 현재 PhoneGap 은 Android, iPhone, Blackberry, Nokia 와 같은 대부분의 플랫폼을 지원하고 있다. PhoneGap 은 Javascript 프로그래밍 인터페이스를 사용하여, 응용프로그램이 가져야 하는 기능들을 Javascript/HTML5 를 통해 기술 한다. 이렇게 작성된 웹 기반 응용프로그램에 WebKit 이라고 불리는 웹 응용프로그램 접근 도구를 사용하여 접근한다. 이를 통하여 PhoneGap 은 스마트폰 내부에 일종의 웹서버를 운영하게 되고 이에 접근하여 기능을 활용하게 된다. 이 방법이 가지고 있는 단점으로는 WebKit 을 이용한 응용프로그램 개발 시 디버깅(Debugging)이 불가능하고, 전용 개발 환경을 통해 개발한 응용프로그램과는 조금 다른 모습을 가진다는 것이다.

Titanium 은 웹 기반 기술을 사용하여 크로스 플랫폼을 지원한다는 점에서 PhoneGap 과 유사하다. 하지만 PhoneGap 은 WebKit 을 사용하여 Javascript 로 개발된 응용프로그램을 를 단순히 해석하여 보여주는데 그치는 반면, Tatanium 은 Mozilla Javascript Engine 을 통해 스마트폰이 제공하는 라이브러리들과의 응집력을 강화 시켰다. 이를 통하여 Tatanium 을 이용하여 개발된 응용프로그램은 실제 스마트폰 응용프로그램과 비슷한 결과물을 만들어 낼 수 있다.

Appspresso 는 국내 KTH 에 의해서 개발된 크로스 스마트 플랫폼 프레임워크이다. Appspresso 는 앞서 언급한 PhoneGap, Titanium 과 마찬가지로 웹 기술에 기반한 프레임워크이다. 현재 안드로이드와 iOS 를 지원하고 있다. 내부적인 디자인 결과물이 많이 공개되어 있지 않지만, PhoneGap 과 유사한 방법을 통하여 크로스 플랫폼 개발을 지원하고 있다. Appspresso 는 Waikiki API 라고 불리는 Wholesale Application Community 에 접근하기 위한 장치 API 를 지원한다는 점에서 기존의 프레임워크들과는 차별성을 가진다.

Corona 는 앞서 언급한 PhoneGap, Tatanium, Appspresso 와는 전혀 다른 방법을 통해 크로스 플랫폼 개발을 지원하고 있다. Corona 는 Lua 스크립트 프로그래밍 언어를 사용하여 개발된 응용프로그램을 실제 스마트폰에서 실행 가능한 코드로 변환하는 방법을 사용하고 있다. Corona 는 풍부한 그래픽 기반의 개발을 지원하는데 초점을 맞추고 있다. 또한 디버깅 도구와 시뮬레이터를 지원하고 있어 기존의 프레임워크와는 차별성을 가지고 있다. 기존의 웹기반 크로스 플랫폼 개발 도구의 경우 응용프로그램의 기능을 지원하는데 장점을 가졌다면, Corona 의 경우는 2D 게임과 같은 응용프로그램 개발에 장점을 가진다. 현재 Corona 는 안드로이드와 iOS 를 지원하고 있다.

3. 스마트폰 플랫폼의 공통점 분석

다양한 스마트폰 플랫폼을 지원하는 통합개발 환경을 구축하기 위해서는 최우선적으로 현재 존재하는 스마트폰 플랫폼들의 분석이 필요하다. 분석대상은 현재 스마폰 세계 시장의 대부분을 점유하고 있는 구글

의 안드로이드와 애플의 iOS 를 대상으로 수행하였다.

3.1. 구글 안드로이드 플랫폼 분석

안드로이드 플랫폼은 리눅스 커널, 라이브러리, 안드로이드 런타임, 응용프로그램 프레임워크, 그리고 응용프로그램 계층으로 구성된다[3]. 가장 하위 단계에 위치한 리눅스 커널은 하드웨어와 소프트웨어를 연결하는 역할을 담당하며 보안, 메모리 관리, 프로세스 관리, 네트워크 관리 등을 수행한다. 응용프로그램 프레임워크 계층은 개방형 개발 플랫폼을 지원하는 것을 목표로 하며 위치 정보 접근, 백그라운드 서비스 지원, 알람 설정, 알림 추가와 같은 다양한 기능들을 제공한다. 이를 통해 개발자는 응용프로그램들을 더욱 손쉽게 개발 할 수 있다. 마지막으로 응용프로그램 계층은 안드로이드 플랫폼에서 기본적으로 제공하는 문자, 전화, 지도, 달력과 같은 응용프로그램과 사용자가 개발한 새로운 응용프로그램들을 지칭하며 하위 계층인 안드로이드 프레임워크를 이용하여 시스템 및 장치와 소통한다.

스마트폰을 위한 응용프로그램 개발에서 가장 중요한 부분은 사용자 인터페이스의 개발이다. 사용자 인터페이스는 사용자와 응용프로그램 사이에 어떠한 방식으로 정보를 주고받을지를 정의하는 중요한 역할을 한다. 특히 스마트폰 응용프로그램의 경우 자체적인 정보 처리보다는 정보 접근과 표현이 중요하기 때문에 사용자 인터페이스의 개발이 매우 중요하다고 할 수 있다. 안드로이드의 사용자 인터페이스는 Vie w 클래스와 ViewGroup 클래스를 이용하여 구현된다. View 클래스는 안드로이드 플랫폼에서 사용자 인터페이스를 구현하는데 사용되는 가장 기본적인 단위이다. View 클래스의 하위에는 버튼, 텍스트 필드와 같이 미리 구현되어 제공되는 위젯(widget)들이 존재한다. 또한, ViewGroup 클래스는 테이블 형식, 배열 형식과 같은 사용자 인터페이스 내에 위치하는 Veiw 들의 관계를 정하는 레이아웃을 개발하는데 사용된다. View 클래스와 ViewGroup 클래스는 서로 참조가 가능하며 서로의 하위에 존재 할 수 있다. 그림 1 는 View 클래스와 ViewGroup 클래스 사이의 관계를 나타내고 있다. 이러한 클래스의 정의에 따라 메모리가 할당되고 처리 가능한 형태가 되면 객체로 지칭한다. 안드로이드 플랫폼에서 사용자 인터페이스를 그리기 시작하는 것은 View 클래스 또는 ViewGroup 클래스의 최상위 객체로부터 시작된다. 최상위 객체는 자식 객체들에게 필요한 정보와 계산을 수행하도록 요청한다. 그리기 위한 계산이 완료되었을 경우, View 객체와 ViewGroup 객체을 탐색하면서 사용자 인터페이스를 그리기 시작한다. 이 경우, 각 객체들은 자신의 자식 노들을 그리는 것에 대한 모든 책임을 위임 받는다. 이러한 방식으로 인해서 객체들은 부모-자신-자식의 순으로 그려지게 된다.

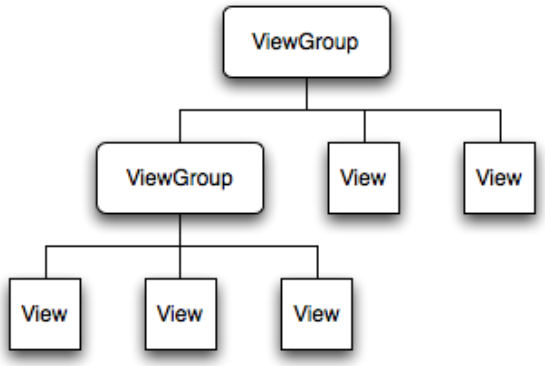

그림 1. View 및 ViewGroup의 관계 [3]

3.2. 애플 iOS 플랫폼 분석

iOS 의 소프트웨어 개발 도구(Software Development Toolkit; SDK)는 응용프로그램 개발, 설치, 실행, 검증에 필요한 도구와 인터페이스를 제공한다 [4]. 응용프로그램들은 iOS 시스템에서 제공하는 각종 프레임워크와 Objective-C 언어를 통해서 개발 할 수 있으며 iOS 가 구동 중인 장치에서 바로 실행 할 수 있다. 또한, iOS 의 장점 중 하나는 사용자의 컴퓨터에 설치되어 있는 iTunes 소프트웨어를 이용하여 사용자의 정보 및 데이터를 저장, 복구, 그리고 다른 장치와 동기화 할 수 있다.

iOS 의 시스템 아키텍쳐는 그림 2 와 같다. 최하위 단계에 위치한 CoreOS 는 시스템 커널을 포함하고 있으며 파일 시스템, 네트워킹, 보안, 전력관리, 장치관리 등을 수행한다. 다음 단계인 Core Service 에서는 장치를 관리 하고 유지하기 위해 필수적인 문자열 관리, 네트워킹, URL 도구, 연락처 관리, 사용자 설정과 같은 프레임워크들을 제공한다. 또한 하드웨어와 연계하여 제공되는 서비스인 GPS, 나침반, 가속도계, 자이로스코프 센서 등의 서비스도 제공한다. Core Service 단은 Foundation 프레임워크를 포함하고 있는데,

이는 응용프로그램 개발에 필요한 데이터 타입의 선언 및 관리 등과 같은 개발에 필수적인 자료를 제공한다. 다음 단인 Media 은 Core Service 단에서 제공하는 프레임워크를 이용하고 그래픽 요소와 멀티미디어 서비스를 상위 단인 Cocoa Touch 에 제공한다. Cocoa Touch 단은 iOS 에 기반한 응용프로그램들은 직접적으로 지원하는 프레임워크이며 개발의 편의를 위해 미리 개발된 게임 도구, 지도 도구, 광고 도구와 같은 다양한 함수 및 도구들을 제공한다. Cocoa Touch 단은 Objective-C 에 기반하여 제작되었으며 개발자가 개발하는 환경과 동일하다. Cocoa Touch 단에 포함되어 있는 UIKit 은 사용자 인터페이스를 개발하고 정의 하는데 필요한 자료 구조 및 지원 도구들을 포함하고 있어 매우 중요하다.

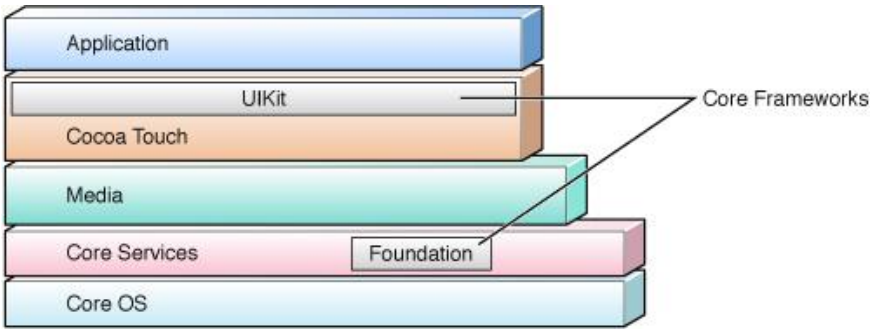

그림 2. iOS의 아키텍처 [5]

iOS 에서 사용자 인터페이스를 구현하기 위해서는 View controller 와 UIView 클래스 사이의 관계에 대해서 이해하는 것이 매우 중요하다. View controller 클래스는 자신에게 속한 UIView 객체들을 관리하고 그들이 가지고 있는 컨텐츠를 적절하게 표현하는 것을 담당한다. 하지만, UIView 객체들을 관리하는 것만이 View controller 의 역할은 아니며, 대부분의 View controller 객체들은 다른 객체와 통신하고 전환이 일어나야 할 시기에 수행해야 할 작업들을 조율한다. View controller 클래스는 MVC 소프트웨어 디자인 패턴에 기반하여 설계 되었다. UIView 객체는 응용프로그램 윈도우 내에서 특정 사각 영역을 관리하는 역할을 하고, 사용자 인터페이스를 구성하는 가장 기본적인 블록이다. UIView 클래스에는 컨텐츠 표현, 사용자 이벤트 처리, 그리고 하위 UIView 객체들의 레이아웃을 관리하는 역할이 할당되어있다. UIView 객체들은 계층 구조를 이룰 수 있으며, 상위 view 객체는 하위 UIView 객체들의 위치와 크기를 조정하거나 조율하는 역할을 수행한다. 그림 3 는 Window, View controller, 그리고 UIView 클래스 사이의 관계를 보여준다.

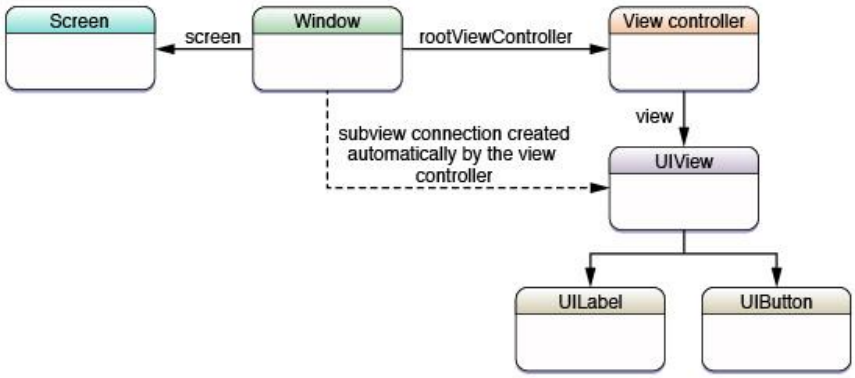

그림 3. iOS에서 View controller와 View 객체 사이의 관계 [6]

4. 플랫폼 독립적인 정보 모델 설계

정보 모델은 특정 도메인 내에 존재 하는 개념, 관계, 제약사항 들을 표현하기 위해 사용된다. 정보 모델을 활용하여 특정 도메인 내에 존재하는 지식들을 표현하고 이를 활용한 정책, 추론, 그리고 학습 기술들을 활용한 기술들이 활발히 연구 중에 있다 [7,8]. 도메인 내에 존재하는 지식을 포함한 정보 모델은 급격하게 변하는 요구사항과 환경에 대응하기 위해서 자동적으로 시스템 내에 존재하는 구성요소들을 조율하는데 사용될 수 있다. 정보 모델의 가장 큰 장점은 개발에 관련된 대부분의 지식을 표현 할 수 있으며 특정 기술의 제작자가 제공하는 지식에 의존하지 않은 채 기술들을 표현 할 수 있다는 데 있다. 따라서 특정 기술 및 플랫폼에 의존하지 않고 지식 및 응용프로그램을 표현하고 이를 바탕으로 실행가능 한 코드를

생성하기 위한 목적으로 사용되기에 최적의 특성들을 가지고 있다.

정보 모델은 일반적으로 Unified Modeling Language(UML)을 사용하여 표현된다. UML 은 표준화된 일반 모델링 언어로써 객체지향 소프트웨어 공학 분야에서 객체들의 관계 및 제약 사항들을 표현하기 위해서 주로 사용된다. 모델 기반 소프트웨어 아키텍처 설계 방법에서는 UML 모델을 활용하여 실행 플랫폼에 독립적으로 응용프로그램을 명세하고 이를 바탕으로 실제 플랫폼에서 실행 가능한 코드를 생성하는 방법을 핵심으로 하고 있다.

앞서 3 절에서는 구글의 안드로이드와 애플의 iOS 플랫폼에 대한 분석을 수행하였다. 이 분석 결과를 기반으로 각 플랫폼 사이에 존재하는 공통점을 추출 할 수 있다. 안드로이드와 iOS 의 분석결과를 살펴 보면, 두 개의 플랫폼은 상당히 유사한 방식으로 사용자 인터페이스 설계를 지원하고 관리하도록 구현되어 있다는 것을 알 수 있다. 두 개의 플랫폼 모두 View 객체를 이용하여 컨텐츠를 관리하고 그들을 표현하도록 하고 있으며, 각 객체는 모두 주어진 window 내에서 사각형의 영역 내에서 발생하는 이벤트를 처리하도록 역할을 위임하고 있다. 이러한 기본적인 공통점을 바탕으로 정보 모델(Information Model)을 설계하여 두 플랫폼 모두에 적용 가능한 사용자 인터페이스를 설계를 명세할 수 있다. 이렇게 생성된 정보모델은 다음 개발 과정에서 코드 생성에 사용된다.

그림 4 는 각 플랫폼의 공통적인 부분을 추출하여 설계한 정보 모델의 최상위 부분을 나타내고 있다. 각 응용프로그램은 하나의 윈도우를 가지고 있으며, 윈도우 내에는 View 또는 View 의 관리를 위한 ViewController 가 있다. View 와 ViewController 는 모두 ViewContainer 의 하위 클래스 이며, 각자는 새로운 View 객체 또는 ViewController 객체를 가질 수 있다. 여기서 ViewController 는 안드로이드 플랫폼에서는 ViewGroup 클래스, iOS 에서는 View controller 클래스에 대응하는 개념이다. 각 ViewController 는 관심 있는 Event 들을 등록 할 수 있고, 만약 Event 가 발생시 적절한 Action 을 수행 할 수 있다. Event 객체는 사용자의 제스쳐, 터치 모션, 타이머, 콜백(Callback) 함수 들을 지칭하는 개념으로 응용프로그램의 상태변화를 시작하는데 사용된다. 특정 Event 가 발생 시 각 View 는 Event 를 처리하기 위한 Action 을 실행하게 된다. Action 은 미리 정의된 View 의 상태 변화를 표현하기 위한 클래스로 View 의 컨텐츠를 바꾸거나 특정 동작을 수행 할 수 있다. 이 과정에서 각 Action 들은 수행하기 위한 조건을 충족했는지를 Condition 을 참조하여 알 수 있다. 이 일련의 과정을 예로 들면, 사용자가 버튼(View)을 눌렀을 경우(Event), 버튼을 사라지게 하는 동작(Action)을 하는 경우를 생각할 수 있다.

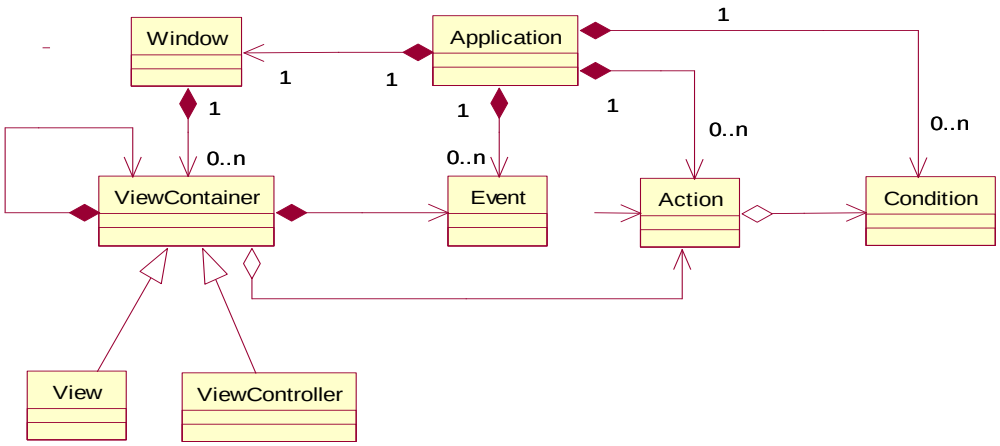

그림 4. 플랫폼 독립적인 정보모델의 일부

그림 5 는 ViewContainer 의 하위 계층을 도식화한 것 이다. View 의 하위 계층으로 Widget 과 UserDefinedView 가 있다. Widget 은 각 플랫폼에서 이미 개발되어 제공되는 View 를 말하는 것으로 대표적으로 버튼, 텍스트, 라벨, 이미지와 같은 것들이 있다. UserDefinedView 는 Widget 또는 Graphic 도구를 이용하여 개발자가 직접 설계한 Veiw 를 지칭한다. ViewController 클래스는 하위에 속한 View 들을 관리하고 그들이 사이에 관계를 조율한다. 여기에 속한 대표적인 클래스로는 Layout 들이 있다. Layout 들은 주어진 영역 내에서 각각의 View 들이 어떻게 배치되어야 하는지를 설정하는 역할을 수행한다.

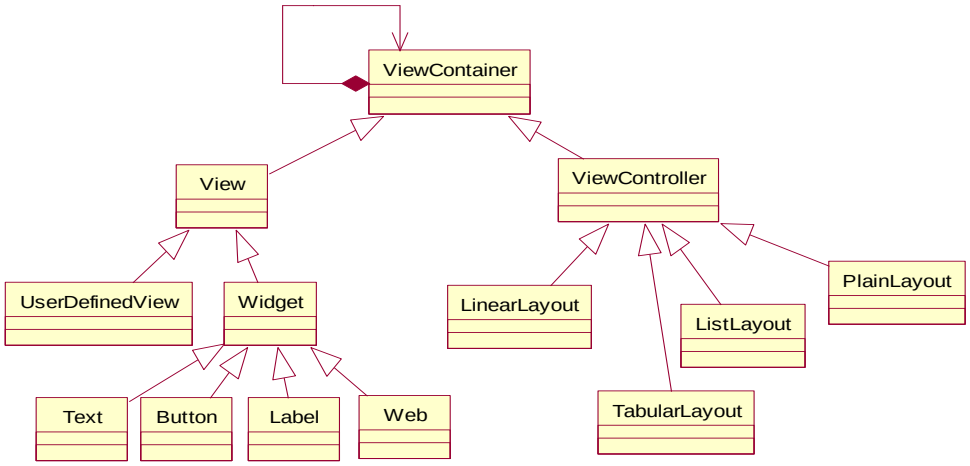

그림 5. ViewContiner의 계층 구조의 일부

그림 6 은 Event 와 Action 클래스의 하위 계층들을 UML 을 사용하여 표현한 것이다. Event 클래스는 UserEvent 와 SystemEvent 로 세분화 될 수 있는데, UserEvent 는 사용자가 입력하는 이벤트를 지칭하며 SystemEvent 는 Kernel 으로부터 전달되는 이벤트들을 말한다. UserEvent 의 종류로는 사용자가 화면을 터치, 누르기, 멀티 터치 등의 이벤트를 생각할 수 있다. SysetmEvent 는 운영체제로부터 응용프로그램에 전달되는 이벤트이며 타임아웃, 응용프로그램 종료 요청 등이 있다. 이벤트를 수신 시 이를 처리과정을 명세하기 위해 Action 클래스가 존재한다. Action 클래스는 미리 정의된 몇 가지 행동을 수행 할 수 있다. Action 클래스의 대표적인 예제로는 View 의 숨김/보임등이 있다.

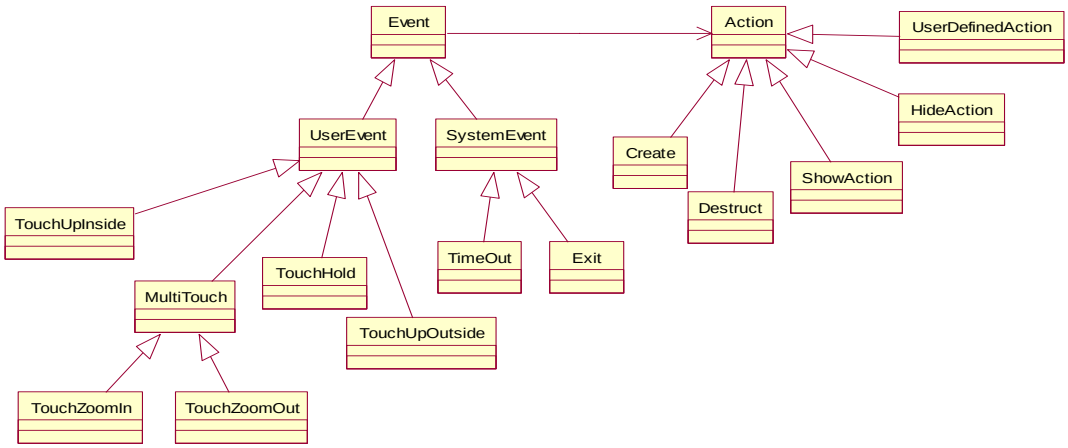

그림 6. Event와 Action 모델의 일부

5. 응용프로그램 개발 프로세스

통합 개발을 위한 프로세스는 4 절에서 개발한 정보모델에 기반하여 진행된다. 플랫폼 독립적으로 명세된 응용프로그램은 그림 7 과 같은 3 단계를 거쳐 실행 가능한 형태로 바뀌게 된다. 개발 프로세스는 크게 장치 독립적 명세(Device Neutral Implementation; DNI), 모델 기반의 변환(Model Based Translation; MBT), 그리고 장치 종속적 명세(Device Specific Implementation; DSI)의 3 단계의 과정을 거치게 된다.

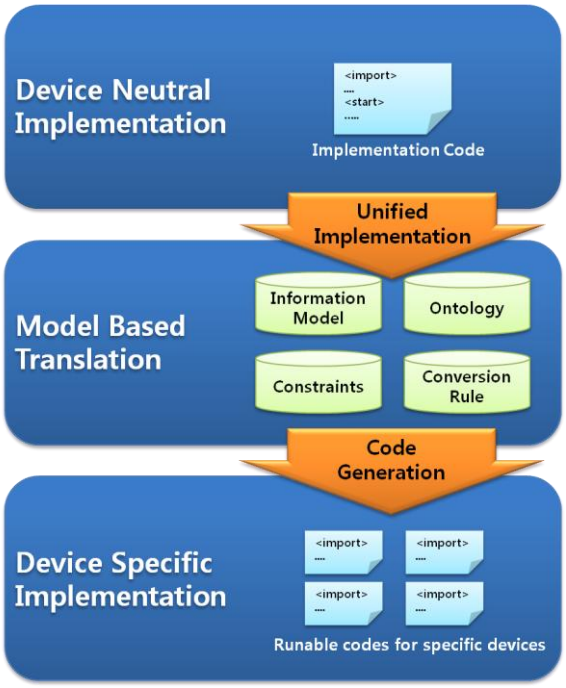

그림 7. 통합 환경을 이용한 개발 프로세스

DNI 단계에서는 스마트폰 플랫폼에 의존적이지 않은 독립적인 방법과 도구를 통하여 개발하려고 하는 응용프로그램의 기능과 사용자 인터페이스를 명세 한다. 이 과정은 4 절에서 작성한 정보 모델에 기반하여 응용프로그램을 명세 할 수 있다. 4 절에서 설계한 정보모델은 일종의 스키마(Schema)로써 각 플랫폼에 존재하는 개념들을 표현한 것이다.

다음 단계인 MBT 에서는 정보 모델과 통합 개발 도구를 사용하여 명세 된 응용프로그램을 장치 종속적인 모델로 변화한다. 이전 단계의 정보 모델은 특정 장치나 플랫폼에 독립적으로 응용프로그램의 기능 및 사용자 인터페이스만을 명세 했기 때문에 실제로 구동 가능한 응용프로그램을 대체하는 모델은 아니다. 따라서 특정 플랫폼 또는 장치에서 실행하기 위한 코드를 생성하기 위해서는 특정 플랫폼 또는 장치에 기반한 정보 모델이 필요하다. 이 단계는 DNI 단계에서 명세된 응용프로그램은 모델 변환 기법을 활용하여 각 스마트폰 플랫폼에 동일한 의미 및 기능을 제공하는 형태로 변환한다. 이 과정에서 추가적으로 필요한 정보들은 별도로 미리 정의된 규칙 따라 추출 할 수 있다.

개발의 가장 마지막 과정인 DSI 단계에서는 MBT 과정에서 생성된 플랫폼 종속적인 모델을 이용하여 실제 장치에서 구동 가능한 코드를 생성, 검수하고 각 스마트폰 플랫폼의 특성을 살릴 수 있도록 세부 조정 및 추가적인 개발을 한다.

6. 통합환경에서 응용프로그램 명세

플랫폼 독립적인 정보모델을 활용한 스마트폰 응용프로그램 명세 환경을 위해서는 먼저 UML 모델을 표현 할 수 있는 도구가 필요하다. UML 모델을 컴퓨터에서 처리 가능하도록 표현하는 도구로는 IBM 사에서 개발한 Rational Rose 를 사용하였다. Rational Rose 는 컴포넌트 기반의 소프트웨어 개발을 지원하기 위해 개발된 도구로써 다양한 모델링 방법을 지원하고 생성된 모델들을 소프트웨어 개발에 연동할 수 있는 도구를 지원한다 [9]. 그림 8 은 Rational Rose 를 사용하여 UML 을 이용하여 플랫폼 독립적인 정모 모델을 설계하는 모습이다. Rational Rose 를 사용하여 설계된 정보 모델은 응용프로그램을 명세하기 위한 스키마(Schema)의 역할을 하게 된다. 즉, 이 모델은 실제 응용프로그램의 명세를 위한 틀을 제공하는 것이다.

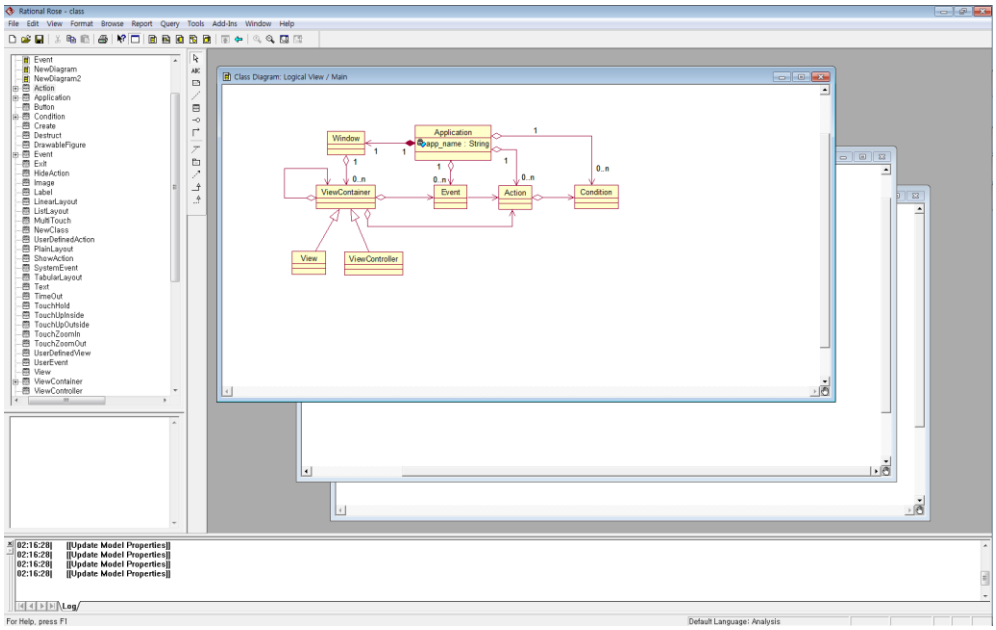

그림 8. Rational Rose를 이용한 UML 모델링

Rational Rose 를 이용하여 설계된 정보 모델은 응용프로그램 통합 명세환경 구축을 위하여 사용된다. 이 과정에서 Eclipse 의 Eclipse Modeling Framework(EMF)를 사용하여 모델 기반의 명세 도구를 쉽게 개발할 수 있다. EMF 는 구조화된 데이터 모델에 기반하여 모델링 프레임워크와 코드 생성 도구를 제공하여 준다. EMF 는 명세된 모델을 실제 응용프로그램과 결합 할 수 있도록 JAVA 코드 생성, 모델의 시각적 표현과 그를 위한 어댑터, 그리고 모델을 생성하고 수정할 수 있는 편집기를 제공하여 준다. 또한 다른 응용프로그램에서 사용되는 모델을 질의하고, 모델 처리, 모델 검증들을 편리하게 할 수 있도록 도와준다 [10]. 그림 9는 이렇게 개발된 응용프로그램 명세도구의 작동 모습을 보여준다.

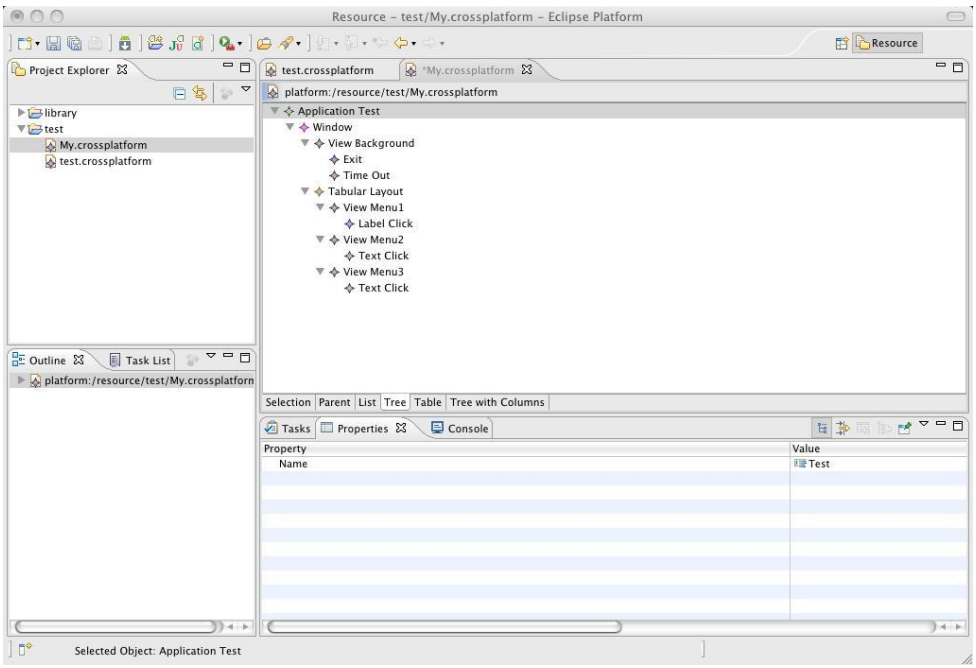

그림 9. 모델을 이용한 응용프로그램 명세 환경

7. 플랫폼 종속적 정모 모델 변환 및 코드 생성

이전 단계에서 명세한 스마트폰 응용프로그램은 어느 플랫폼에도 종속적이지 않은 형태로 표현이 되어 있다. 이 모델을 이용하여 실제 장치에서 수행 가능한 형태를 만들기 위해서는 명세 된 기능들과 기술들이

각 플랫폼에서 사용하고 있는 기술들과 매칭되도록 변환을 해주어야 한다. 이 과정은 플랫폼 독립적인 모델을 플랫폼 종속적인 모델로 변환하는 과정이다. 그림 10 은 플랫폼 독립적인 모델로부터 실행 가능한 소스코드 생성까지의 일련의 변환 과정을 보여준다. 플랫폼 독립적인 모델을 통하여 명세 된 응용프로그램은 모델 변환 기술을 이용하여 플랫폼 종속적인 모델로 바뀌게 된다. 예를 들어 3.4 에서 명세한 View 의 경우 안드로이드 에서는 Anrorid.view 클래스로 iOS 에서는 UIKit 내부의 UiView 클래스로 변환되어야 한다. 이러한 일련의 과정들은 Generative Programming 영역에서 활발히 연구 되고 있다 [11].

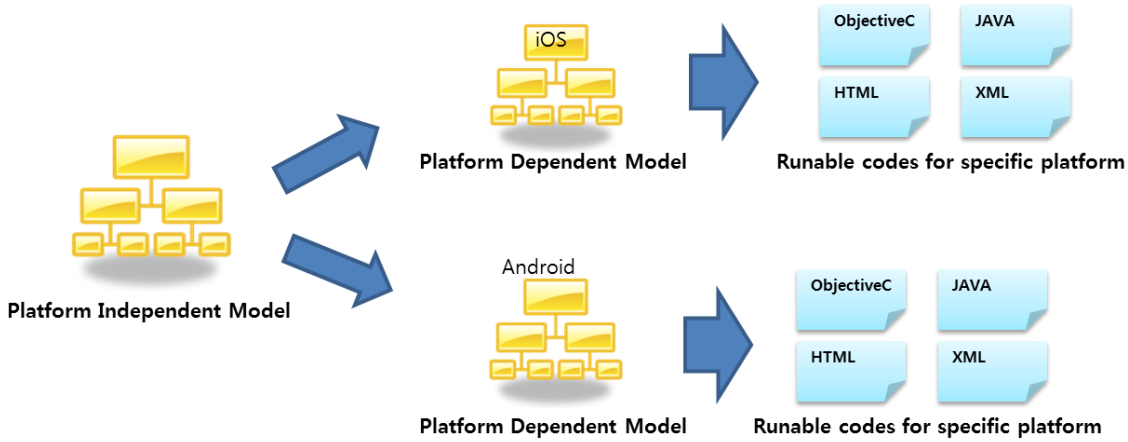

그림 10. 모델 사이의 변환 과정

플랫폼 독립적인 모델에서 플랫폼 종속적인 모델로 변환하는 과정에서는 Atlas Transformation Language(ATL)가 사용이 된다. ATL 은 ATLAS INRIA & LINA 연구그룹에 의해 개발 된 모델 변환 언어이다. 모델 기반의 개발을 지원하기 위해 개발되었으며 변환 규칙들을 명세하여 특정 모델을 다른 모델로 변환할 수 있다 [12]. ATL 은 Eclipse 플랫폼의 플러그인 형태로 개발되었으며 다양한 모델들을 쉽게 변환할 수 있는 도구를 지원하여 준다. 특히 ATL 은 EMF 에서 사용하고 있는 Ecore 모델을 지원한다. ATL 모델 변환 규칙을 명세하여 플랫폼 독립적인 언어를 플랫폼 종속적인 언어로 표현된 Ecore 모델로 변환할 수 있다. 그림 11 는 ATL 변환 규칙의 명세의 예시이다.

```
prefix pre: "/resource/cross_platform"

map package view def = {
        uriPattern= pre + self.name
        properties = {
                map attr name with android.view;
        }
}
```

그림 11. ATL 변환 규칙 예시

마지막으로 코드를 생성하기 위한 과정은 플랫폼 종속적인 모델에서 각 개념을 코드로 변환하여 출력 파일에 작성하게 된다. 코드 변환을 위해서 미리 작성된 템플릿에 각 개념에 해당하는 코드 조각을 작성하여 두고 코드 변환 요청이 왔을 시 이를 참조하여 실행 가능한 코드를 작성하게 된다. 그림 12 은 실행 가능한 코드의 조각을 저장하기 위한 템플릿의 예시이다. 이러한 코드생성 방법은 간단한 코드의 생성은 효율적으로 수행 할 수 있지만, 복잡한 코드의 생성을 위해서는 더욱 많은 연구가 필요하다.

```
Class= android.TextView
Header code= "import android.widget.TextView;"
Location= "onCreate"
Body Code= "TextView " + getID(self.name) +
            " = new TextView(this);₩n" + "tv" + .getID(self.name) +
            "(" + getValue(self.name) +");"
```

그림 22. 코드 조각을 저장하기 위한 템플릿 예시

8. 결론

본 연구의 목적은 다양한 모바일 플랫폼에 적용 가능한 스마트폰 응용프로그램 개발 방법을 제시하고 이를 지원하는 도구의 개발이다. 이를 구현하기 위해 플랫폼 독립적인 정보 모델의 명세를 통한 실행 가능한 코드 생성 방법을 제안하였다. 본 논문에서는 제안된 방법을 통하여 간단한 스마트폰 응용프로그램을 개발 할 수 있는 가능성을 보였다. 본 연구는 스마트 폰 앱의 개발 시간과 비용을 단축시키는데 중점을 두고 있다. 이를 이용하여 스마트 폰 앱 개발자들이 기존의 각 플랫폼에 맞는 새로운 앱을 개발 하는데 낭비하였던 시간 및 비용을 절감 할 수 있을 것으로 기대된다.

현재 본 연구에서 제안된 방법만을 멀티미디어의 제어와 같은 복잡한 논리 처리가 요구되는 응용프로그램을 개발하는 것은 아직까지는 불가능하다. 따라서 이러한 문제를 극복하기 위하여 플랫폼 독립적 정보 모델의 확장을 통한 더욱 다양한 논리와 기능 명세 방식의 제공, 각 플랫폼의 특징을 수용할 수 있는 정보 모델의 개발, 온톨로지 모델과의 연동을 통한 명세된 응용프로그램의 검증, 많은 사례 연구를 통하여 제안된 방법의 실효성 검증와 같은 향후 연구가 필요하다.

9. 참고 문헌

[1] S. Allen, V. Graupera and L. Lundrigan. "The Smartphone is the New PC," PRO SMARTPHONE CROSS-PLATFORM DEVELOPMENT, pp. 1-14, 2010.

[2] A. M. Chris. "Bridging the Mobile App Gap," Connectivity and the User Experience, 2011.

[3] Google Inc., "The Developer's Guide," March 23, 2012, Available at http://developer.android.com/guide/index.html.

[4] Apple Inc, "iOS Technology Overview," October 10, 2011, Available at https://developer.apple.com/library/ios.

[5] Apple Inc., "Cocoa Fundamentals Guide," December 13, 2010, Available at https://developer.apple.com/library/ios.

[6] Apple Inc., "View Controller Programming Guide for iOS," Jenuary 16, 2012, Available at https://developer.apple.com/library/ios.

[7] B. Jennings, S. van der Meer, S. Balasubramaniam, D. Botvich, M. Ó Foghlú, W. Donnelly and J. Strassenr, "Towards Autonomic Management of Communications Networks," IEEE Communications Magazine, Volume 45, Issue 10, October 2007.

[8] J. Strassner, N. Agoulmine, and E. Lehtihet " FOCALE: A Novel Autonomic Networking Architecture," Latin American Autonomic Computing Symposium, Campo Grande, MS, Brazil, Jul. 18-19, 2006.

[9] Rational Software Corporation, "Rational Rose 2000e Rose Extensibility User's Guide," March 2000.

[10] F. Budinsky, D. Steinberg, E. Merks, R. Ellersick and T. J. Grose, "Eclipse Modeling Framework," Addison-Wesley Professional, August 21, 2003

[11] D. Agrawal, T. Levendovszky, J. Sprinkle, F. Shi , G. Karsai, "Generative programming via graph transformations in the model-driven architecture," OOPSLA Workshop, 2002.

[12] ATLAS group, LINA & INRIA, Nantes, "ATL: Atlas Transformation Language User Manual," February 2006.

한 윤 선

2009 포항공과대학교, 컴퓨터공학과 학사

2009 ~ 현재 포항공과대학교, 정보전자융합공학부 석/박사 통합 과정

<관심분야> 상황인지 서비스, 정보 모델링, 자율 컴퓨팅

홍 원 기

1983 Univ. of Western Ontario, BSc in Computer Science

1985 Univ. of Western Ontario, MS in Computer Science

1985 ~ 1986 Univ. of Western Ontario, Lecturer

1986 ~ 1991 Univ. of Waterloo, PhD in Computer Science

1991 ~ 1992 Univ. of Waterloo, Post-Doc Fellow

1992 ~ 1995 Univ. of Western Ontario, 연구교수

1995 ~현재 포항공과대학교 컴퓨터공학과 교수

2007~2011 포항공과대학교 정보통신대학원장

2007~2010 포항공과대학교 정보통신연구소 연구소장

2008~2010 포항공과대학교 컴퓨터공학과 주임교수

2008~2012 포항공과대학교 정보전자융합공학부장

2008 ~현재 포항공과대학교 정보전자융합공학부 교수