모듈 방식을 이용한 안드로이드 앱 개발 시스템

박주연[†], 박성모^{††}

Android App Development System Using Modular Method

Ju-Yeon Park[†], Seong-Mo Park^{††}

ABSTRACT

As the more people use smart phones, the interest in the apps gets the higher. Studies such as App Inventor, app generation methods using templates provide app development process with app development methods by substituting programming work. However, the realm of producible apps is limited and there are a lot of set up operations and input informations. Also, there is lack of support for smart phone sensors that are in a high demand of utilization. This paper proposed an android app development method for resolving existing problems, and implemented an accompanying app development tool. When the proposed app development method derived through combination of function modules and sensor modules is used, it's possible to produce apps with minimal user inputs, and to use sensors easily. Also, because it is simple to identify overall flow of app execution, and functions addible to the app are provided for users by units of module, it is possible to develop apps quickly.

Key words: Android, Smart phone, App, Development System, Modular Method, Sensor

1. 서 론

스마트폰의 발전과 함께 국내의 스마트폰 사용자는 2013년 6월 기준으로 3천 5백만 명을 넘어서고 있다[1]. 앱 스토어를 통해서 스마트폰에 설치할 수 있는 다양한 앱들은 사람들이 스마트폰을 선택하는 중요한 이유 중 하나이다[2]. 앱 스토어에 등록된 앱들은 게임, 정보, 엔터테인먼트 등 다양한 분야가 존재하며 스마트폰 사용자는 원하는 분야의 앱을 다운로드 받아 사용할 수 있다.

앱은 스마트폰에 탑재된 운영체제의 제작사에서 배포하는 SDK(Software Development Kit)를 이용해 만들어진다. 이는 전문 개발자를 위해 제공되는 방법으로써 앱 스토어 자체는 누구나 앱을 만들어

등록할 수 있게 개방되어 있지만 일반 사용자는 앱을 개발하는 것 자체가 어려운 실정이다. 스마트폰을 사용하는 사람이 많아지고 앱에 대한 관심이 증가함에 따라 프로그래밍 없이 앱을 제작하는 방법과 도구 개발에 대한 연구가 진행되고 있으며 현재 사용할 수 있는 여러 앱 제작 도구들이 출시되어 있다.

일반적으로 앱 개발 과정에서는 알고리즘, 자료 구조, 앱 프로세스의 처리 흐름, 프로그래밍 언어 등의 이해를 필요로 한다. 이것은 앱을 개발할 때 먼저 학습되어야 하는 부분으로 개발자가 아닌 사람에게 앱 개발을 어렵게 하는 부분이다. 또한 센서를 사용하려고 할 때는 기능 구현에 센서 제어 기술 및 프로그래밍을 필요로 하기 때문에 앱 제작은 더욱 어려워진다. 하지만 프로그래밍을 배우지 않은 일반 스마트

※ Corresponding Author: Seong-Mo Park, Address: (500-757) Yongbong-ro 77, Buk-gu, Gwangju, Korea, TEL: +82-62-530-1798, FAX: +82-62-530-3439, E-mail: smpark@jnu.ac.kr

Receipt date: Nov. 19, 2013, Revision date: Jan. 21, 2014
Approval date: Apr. 7, 2014

^{††} Dept of Electronics and Computer Engineering, Chonnam National University
(E-mail: lindolsang@nate.com)

^{††} School of Electronics and Computer Engineering, Chonnam National University
(E-mail: smpark@jnu.ac.kr)

폰 사용자들도 앱을 만들고 싶을 경우에는 프로그래밍 없이 쉽게 앱을 제작할 수 있는 도구의 도움을 받는다면 앱 제작에 대한 흥미와 자신감을 얻을 수 있다[3]. 따라서 프로그래밍 없이 스마트폰에 부착된 다양한 센서들과 API(Application Programming Interface)의 기본 기능들을 사용하는 앱을 쉽게 제작할 수 있으며 다른 사람에게 전송할 수 있는 방법에 대한 연구가 필요하다.

본 논문에서는 기존 연구를 개선하기 위한 안드로이드 앱 생성 방법을 제안하고 앱 개발 툴로 구현하였다. 제안된 기능 모듈과 센서 모듈의 조합을 통한 앱 개발 방법을 사용하면 최소한의 사용자 입력을 통해서 앱을 제작할 수 있으며 센서를 쉽게 사용할 수 있다. 또한 앱이 실행되는 전체 흐름을 파악하기 쉽고 사용할 기능들이 모듈 단위로 제공되기 때문에 앱을 빠르게 개발할 수 있다.

본 논문의 구성은 다음과 같다. 2장에서는 앱 개발과 관련된 기존의 연구들에 대한 분석과 이에 대한 보완 방법을 알아보고, 3장에서는 앱 제작을 위해 제안하는 방법에 대해 설명한다. 4장에서는 제안된 방법을 구현해 기존 툴과 비교를 통해 분석한다. 5장에서는 결론을 맺는다.

2. 관련 연구

안드로이드의 제작사인 구글을 통해 제공되는 Android SDK(Software Development Kit)는 앱을 제작하는 가장 강력하고 전문적인 개발 방법이다. 이 방법은 Eclipse에 ADT Plugin과 Android SDK를 설치해야 한다. 또한 앱을 제작할 때는 안드로이드 시스템의 구성 요소와 전체 개발 Workflow를 이해해야 하며 Java언어로 Android SDK의 API를 사용하는 프로그램 코드를 작성해야 한다. Android SDK는 주로 전문적인 개발 지식을 갖춘 개발자들이 이용하

여 앱을 개발한다. 하지만 실제 안드로이드 앱을 다운로드 받고 사용하는 주체는 개발과는 거리가 먼 프로그래밍을 알지 못하는 일반 사용자들이다. 비개발자인 사용자가 프로그래밍과 같은 개발 지식을 갖추는 것은 어려운 일이기 때문에 사용자가 앱 제작에 대한 관심을 갖고 기본으로 제공되는 Android SDK를 사용해서 앱을 제작하는 것은 어렵다. 따라서 일반 스마트폰 사용자들의 앱 제작을 돕기 위해 앱 제작과정에서 프로그래밍이 필요 없는 앱 제작 도구가 개발되어 왔다. Table 1은 대표적인 앱 제작 도구들과 이에 대한 비교 내용이다.

App Inventor는 미리 제공되는 컴포넌트 단위의 구성 요소들을 화면에 위지윅 방식으로 배치한다. 배치된 구성 요소들은 블록 에디터(Block Editor)를 통해서 화면 내에서 구성 요소들이 어떻게 동작할 것인지에 대한 논리를 만들어야 한다[4]. 이 과정을 거치게 되면 최종적으로 앱이 완성된다. App Inventor는 앱 제작에서 블록 에디터를 통해 프로그래밍 작업을 없앤 것이 특징이다. 하지만 많은 구성 요소들을 블록으로 변환하고 각 블록간의 논리를 사용자가 조립해야 한다는 점에서는 기존의 프로그래밍 방법과 유사한 면을 가지고 있다. 블록 에디터를 통해 논리를 지정하고 나면 제작된 앱은 APK 파일로 컴퓨터에 저장하거나 PC에 연결된 스마트폰으로 다운로드 할 수 있다. 추가적으로 바코드를 통한 다운로드 방법을 제공한다.

appCookr는 기본으로 제공되는 11가지 레이아웃 템플릿을 통해 화면을 구성하고 사용자가 여기에 모바일 웹 서비스 페이지를 만들어 연결하는 방법을 통해 앱이 만들어진다[5]. 모바일 웹 서비스 페이지는 사용자가 앱을 통해 보여주고자 하는 정보가 포함되어 있다. 그래서 자신이 제공하고자 하는 정보에 대한 부분을 웹 게시물 형태로 만들고 게시물의 URL을 레이아웃에 연결시키면 완성이 되는 구조이

Table 1. Comparison of app development tools

Tool	Execution Type	Pre-installed software	Sensor	Programming	Feature
App Inventor	Native	None	Support	Scratch, Block Editor	Building block
appCookr	Hybrid	None	Not support	Page Link	Mobile web
m-BizMaker	Virtual Machine	m-BizEngine	Not support	Graphic Editor, Setting Property	Database, Biz App

다. 이 방법은 사용자가 앱이 실행되는 환경에 대한 이해가 없이도 앱 제작을 빠르게 진행할 수 있다. 제작된 앱은 사용자에게 APK파일로 제공되며 내부 구현 코드는 사용자가 알 수 없다.

m-BizMaker는 데이터베이스에 대한 처리 부분이 특화되어 있어서 업무용 모바일 애플리케이션을 쉽게 개발할 수 있다. 앱 개발을 위해서는 데이터베이스 처리에 대한 이해가 필요하고 앱을 제작하는 사용자가 기존에 자신이 수행 중인 업무를 앱으로 개발해 처리할 수 있도록 하는 것에 초점을 맞추고 있다[6]. 그리고 m-BizMaker로 제작된 앱은 m-BizEngine이 설치된 IOS와 Android 환경에서 실행된다.

앱 제작 도구를 사용하면 프로그래밍을 모르는 사용자도 Android SDK를 사용할 때 보다 앱을 쉽게 개발할 수 있다. 하지만 앱을 제작하는 경우는 유용하지만 다수가 아닌 특정 사용자를 위한 앱을 제작하고 전달하고자 할 때는 기반 환경을 먼저 설치하고 직접 APK 파일을 복사해야 한다. 또한 기존에 개발된 앱 제작 도구로는 스마트폰과 사용자와의 인터랙션이 추가된 앱을 제작하는데 어려움이 발생한다. 그 이유는 특정 대상과 스마트폰 사이의 인터랙션을 만들기 위해 스마트폰에 부착된 센서를 사용해야 하지만 기존의 툴들은 이를 지원하지 않거나 사용하기 쉽지 않기 때문이다. 그리고 화면 구성에서는 고정적인 레이아웃 템플릿을 이용하기 때문에 화면 구성에 대한 제약과 사용자가 화면에 배치한 구성 요소들의 세부적인 동작 논리들을 각각 지정해야 하는 불편함이 존재한다. 앱을 제작하는 사용자는 일반적으로 제작된 앱을 다른 사용자에게 전달하기 위해서 앱을 마켓에 등록한 후 이를 다른 사용자와 공유한다. 하지만 공개된 공간인 마켓이 아니라 특정 사용자와 개인간 공유를 원할 경우에는 이 방법은 비효율적이기 때문에 앱을 제작한 사용자가 특정 사용자를 지정하면 제작된 앱이 바로 전달될 수 있는 환경이 필요하다.

본 논문에서는 다른 사용자와의 의도하는 시나리오를 앱으로 만들고 이를 지정된 사용자에게 전달할 수 있는 시스템을 통해 기존 앱 제작 방법을 보완한다. 이를 위해 화면 구성은 위지윅 방법을 통해 구성할 수 있고, 스마트폰의 센서는 사용자의 행동이 센서를 통해 인식되는 행동을 단위로 정형화 한다. 또

한 앱을 제작하는 단계는 최소화 하는 안드로이드 앱 제작 시스템과 제작된 앱을 지정된 사용자에게 전달하는 전송 시스템을 설계하고 구현한다.

3. 제안된 앱 제작 시스템

비개발자를 위한 앱 개발 방법에는 비주얼적인 방법으로 앱을 작성하고 작성한 앱의 실행흐름을 쉽게 보여주기 위한 방법이 필요하다. 그 이유는 프로그램 개발에 비주얼적인 요소와 쉬운 문법, 쉬운 오류 수정과정을 제공했을 때 개발 과정의 난이도를 낮출 수 있고[7], 또한 실행 흐름을 시각적으로 보여주는 방법을 제공함으로써 사용자가 프로그램이 실행되는 과정을 쉽게 이해할 수 있기 때문이다[8].

위지윅(WYSIWYG) 에디터를 통한 앱 개발을 지원하기 위해 기능 모듈을 구성하고 센서를 통해 인식되는 행동들의 형태를 정형화 한다. 그리고 기능 모듈이 포함되는 화면들을 단위로 프로그램 실행의 흐름을 표현할 수 있도록 한다.

3.1 앱 제작 시스템 구성

앱 제작 시스템은 사용자가 유저 인터페이스를 통해 제공되는 기능 모듈을 위지윅 에디터로 조합하여 앱을 만드는 시스템이다. 제안된 앱 제작 시스템은 유저 인터페이스를 통해 제공되는 기능 모듈과 센서 모듈을 화면 모듈내에 추가하여 구성하고 모듈과 다른 화면 모듈의 연결 관계를 만들어 앱이 실행되는 논리를 정의한다. Fig. 1은 앱 제작 시스템의 구조와 처리 과정을 나타낸 것이다.

앱 제작 시스템은 유저 인터페이스 상에 3가지 형태의 모듈을 표시하고 위지윅 앱 편집기(GUI 에디터)에서 표시되는 모듈을 가져와 앱이 실행되는 화면을 생성하고 화면 내에서의 동작을 만든다. 사용자가 구성한 화면은 연결 리스트 형태로 에디터에서 관리하며 프로젝트 저장시에 앱 제작에 대한 데이터는 프로젝트 로더(Project Loader)를 통해 XML 형태의 프로젝트 구성파일로 저장하고 참조 리소스(이미지, 음원 등)는 프로젝트 디렉토리로 복사한다. 저장된 프로젝트 구성파일은 앱 제작 시스템의 안드로이드 소스 생성기(Android Source Generator)를 통해 Java언어로 되어 있는 안드로이드 앱 프로젝트 소스를 생성한다. 생성된 소스는 구글의 Android

Fig. 1. Architecture and processing flow of App Development System.

SDK를 통해 최종 빌드 과정을 거쳐 앱 패키지 파일을 생성하게 된다.

3.2 모듈의 정의와 구성 방법

모듈은 사용자가 앱을 만들 때 사용하는 기본 구성 요소로써 위지윅 앱 에디터 상에 UI객체로 표시된다. 모듈은 크게 세 가지 형태로 분류한다. 첫 번째는 앱이 실행될 때 표시되는 한 개의 화면을 나타내면서 다른 모듈을 포함하는 화면 모듈, 두 번째로 사용자가 기능 구현에 사용하는 사용자 기능 모듈, 마지막으로 센서 제어에 사용되는 센서 모듈로 구성된다.

Table 2는 사용자에게 제공 되는 모듈의 종류와 형태에 대한 예이다. 화면 모듈은 스마트폰에서 표시되는 단일 화면을 나타내며 기능 모듈은 사용자가 화면에 추가하여 사용할 수 있는 기능이다. 센서 모듈은 인식하는 기능의 내용에 따라 센서를 통해 다른 화면 모듈로 전환하는 것에 사용한다. 본 논문에서는

Table 2에 제시된 모듈들을 포함하여 기능 모듈 17종, 센서 모듈 5종, 화면 모듈 3종을 사용자가 사용할 수 있도록 하였다.

모듈은 유저 인터페이스를 통해 UI객체로 표시하고 사용자가 사용할 수 있도록 구현한다. Fig. 2는 시스템에서 모듈이 가지는 구성 요소와 구현 내용을 나타낸 것이다. 모듈은 연결화면의 정보, 사용자로부터의 데이터 입력 처리, 사용자의 입력 데이터를 저장하는 부분과 XML 형식으로 모듈이 가진 정보를 저장하는 부분으로 구성된다. 유저 인터페이스를 통해 화면 모듈에 추가된 기능 모듈이 다른 화면 모듈로 전환되도록 설정될 경우 연결 화면 정보에 대상 모듈을 저장(setLinkScreen 함수)한다. 모듈에서 사용자의 설정이 필요한 경우는 사용자 입력 부분에서(selectionPerform 함수) 필요한 데이터를 입력 받도록 코드를 구현한다. 해당 코드를 통해 입력된 정보는 모듈의 데이터로써 저장하고 유저 인터페이스를

Table 2. Examples of the module represented by GUI object

Type	Module Name	Description
Screen Module	Screen Module 1	Screen with title-bar added to editing area
Screen Module	Screen Module 2	Full-screen added to editing area
Function Module	Button Module	Click the button to switch to another Screen Module
Function Module	Image Module	Print an Image
Function Module	Text Input Module	Switch to another Screen Module when user enters the preset text message
Sensor Module	Light Intensity Module	Switch to another Screen Module depending on light intensity
Sensor Module	Turn over Module	Switch to another Screen Module when turn over the smartphone

Fig. 2. Structure and implementation of Module.

표시하고 모듈의 정보를 XML 구성파일로 저장할 때 참조한다. 모듈 내의 구성 파일 부분은(toXml 함수) 시스템에서 제작 중인 앱을 파일 시스템에 저장할 때 프로젝트 로더를 통해 호출되며 함수의 내용은 저장될 정보를 XML 문서로 작성하여 반환하는 코드를 구현한다.

모듈은 기본 API를 사용해 화면에 구성이 가능한 프로그램을 미리 작성해 놓은 코드의 객체이다. 각 모듈의 동작은 독립적이기 때문에 화면에 여러 모듈을 추가하게 되면 각 기능에 맞는 동작이 한 화면 내에서 조합되는 과정을 통해 만들어진다. 기능 모듈은 앱 개발 툴이 실행될 때 플러그인 형태로 로드되기 때문에 구현된 기능 모듈의 수가 늘어날수록 더 다양한 형태의 앱을 만들 수 있다. Fig. 3은 시스템에

서 모듈이 사용자를 통해 사용될 때 관리되는 방법을 나타낸 것이다. 사용자는 유저 인터페이스를 통해 모듈의 추가, 설정, 정보를 입력한다. 유저 인터페이스 부분에서 내부 데이터 모델(Model) 리스트를 통해 사용자가 만든 화면 모듈들을 단위로 모든 모듈 정보를 저장한다. 저장된 모듈 데이터들은 유저 인터페이스의 컨트롤러(Controller)와 뷰(View)를 통해 화면에 표시한다.

Fig. 4는 앨범 커버 이미지를 표시하고 버튼을 눌렀을 때 음악을 재생하는 앱을 제작할 때 이에 대한 모듈이 구성되는 방법을 나타낸다. 앱을 제작하기 위해 Fig. 4의 (a)와 같이 화면 모듈에 기능 모듈들을 추가한다. 그 다음 (a)에서 버튼을 눌렀을 때 보여줄 새 화면 모듈을 생성하고 (b)와 같이 새 화면 모듈에

Fig. 3. Management and use of the module.

Fig. 4. Configuration method of the module. (a) Add Module 1, (b) Add Module 2, (c) App Execution logic control.

음원 재생 모듈을 추가하여 음악이 재생되도록만든다. 최종적으로 (c)와 같이 (a)의 버튼 모듈을 (b)에서 구성한 화면 모듈과 연결을 통해 전체 앱의 동작을 완성한다.

3.3 센서 동작 표현 방법

3.3.1 센서 인식 동작 정의

센서를 사용하기 위해서는 센서 제어 및 데이터 검출 기술을 필요로 한다. 센서를 사용한 사용자의 행동 인식은 스마트폰의 3축 가속도 센서로부터 로그 데이터를 수집하고 데이터에서 주파수 특성을 추출해 동작을 분류하는 과정을 거쳐야 한다[9]. 센서를 통해 인식되는 동작은 물리적 동작과 논리적 동작으로 센서의 종류에 따라 구분한다[10].

비개발자가 센서를 쉽게 사용하기 위해서는 센서 제어나 데이터 검출 과정을 하위에서 처리하되 센서

를 통해 인식되는 물리적 동작을 사용자가 알 수 있는 행동을 단위로 기능을 제공해야 한다. Table 3는 스마트폰의 센서와 인식 행동에 대한 예이다.

3.3.2 센서 모듈의 구현과 사용

스마트폰에 부착된 센서를 통해 인식될 수 있는 행동을 3.3.1절의 설명과 같이 센서에서 인식할 수 있는 물리적 행동으로 분류하고 이를 3.2 절의 모듈 구성 방법으로 센서 모듈을 구현한다. 3.3.1 절에서 정의한 인식 행동에 대해서 센서 모듈 구현시 사용자로부터 입력받아야 하는 정보는 모듈 내부의 사용자 입력부분에 사용자로부터 정보를 입력받는 코드를 구현한다. 구현된 센서 모듈을 화면 모듈에 추가하면 원하는 행동에 대해서 앱에서 사용자의 물리적 행동을 인식하도록 기능을 추가할 수 있다.

다음은 밝기(조도) 센서를 사용자가 사용할 때의 과정이다.

Table 3. Examples of the sensor to recognize the behavior

No	Sensor	Behavior Recognition
1	Light	Bright, Dark
2	Proximity	Near, Far
3	Accelerometer	Shake Smart phone
4	Touch	Touch Screen, Touch three times
5	Geomagnetic	Lean towards east direction

- ① 화면 모듈내에 밝기 센서 모듈 추가
- ② 센서 모듈의 사용자 선택 정보 입력
(밝아질 때, 어두워질 때 중 단일 옵션 선택)
- ③ 센서가 선택한 옵션에 대한 행동을 인식 했을 때 이동할 화면 모듈 연결(지정)

위의 과정을 같이 사용자는 센서 사용을 위한 기능 추가 작업이 간단하다. 실제 센서를 제어하는 코드는 센서 모듈과 사용자가 입력한 모듈 내부의 데이터를 프로젝트 구성파일로 저장하고 난 후 빌더를 통해서 실제 센서 제어 코드가 안드로이드 소스로 생성된다.

3.3.1절의 행동 인식의 예에서처럼 센서를 통해 인식되는 행동을 단위로 기능 모듈을 구현해 놓고 이를 밝기 센서를 사용할 때의 과정처럼 사용자가 모듈을 앱 개발 과정에서 사용하면 비개발자도 센서 기술에 대한 이해 없이 쉽게 센서를 사용하는 것이 가능하다.

3.4 소프트웨어 실행 과정 표시 방법

순서도와 같은 프로그램의 실행 흐름을 표현할 수 있는 방법을 사용하면 제작하는 프로그램의 진행 과정 및 데이터의 변화 과정을 쉽게 이해 할 수 있다[8]. 하지만 순서도를 이용할 경우 프로그램의 실행 흐름을 "Yes", "No" 두 조건에 대해서만 표현할 수 있다. 앱 개발의 경우 다양한 이벤트를 통해 앱의 실행 흐름이 변화하게 됨으로 순서도와 유사한 개념을 표현할 수 있는 트리 구조로 표현하는 것이 적합하다.

트리 구조는 레벨을 통해서 상위와 하위 개념을 표현할 수 있기 때문에 스마트폰 앱에서 이전 버튼을 눌렀을 때 되돌아가는 위치를 명확하게 알 수 있다. 순서도와는 다르게 표현 방법이 두 조건으로 고정되지 않기 때문에 모듈 형식으로 구현된 앱의 기능들이 발생 시키는 화면 전환 이벤트에 대해서 표현이 가능하다. Fig. 5는 본 논문에서 제작한 시스템이 트리 구조로 프로그램의 진행 흐름을 표현하는 방법을 보여준다. 사각형으로 표현된 것은 스마트폰에 표시되는 각 화면을 나타내며 화면 내에 표기된 'S', 'E'는 앱이 최초 시작되는 화면과 종료되는 화면을 의미한다. 각 트리가 의미하는 것은 왼쪽부터 차례로 순차적인 화면 이동(a), 분기 후 이동(b,c), 분기 후 실행 흐름 재결합(d)에 대한 내용이다. 사용자에게 실행 흐름을 시각적으로 보여주기 위해서는 제작하는 앱에 대한 전체 데이터를 트리 형태로 가져야 하며 생

Fig. 5. Representation of the program's execution flow using a tree structure. (a) Sequential go to, (b) go after branch 1, (c) go after branch 2, (d) Combination after branch.

성한 화면들 간의 연결 관계에 대한 정보를 포함해야 한다.

본 논문의 앱 제작 시스템은 사용자가 앱을 제작하는 과정에서 화면 모듈 내에 추가된 기능 모듈과 센서 모듈이 다른 화면 모듈로 전환되는 과정을 인터랙티브하게 시각적으로 보여준다. 이 과정을 통해 사용자는 보다 쉽게 앱이 실행 되는 과정을 바로 바로 확인하면서 앱을 제작 할 수 있다.

4. 구현 및 분석

제안된 기법을 앱 제작 툴로 구현하기 위해 Eclipse Platform의 GEF(Graphical Edit Framework)[11]를 이용했다. Eclipse Platform은 JRE(Java Runtime Environment) 환경 위에서 동작하며 Java 언어를 사용하여 제안된 앱 개발 시스템을 구현하였다.

4.1 제작 툴의 구현

GEF를 이용하여 모듈 데이터들을 트리 형태의 데이터 구조로 만들고 앱 제작을 위한 모듈 편집이 가능한 UI(User Interface)를 구현하였다. 또한 모듈 단위로 기능을 구성하고 클래스화하여 팔레트에 추가하면 되기 때문에 제안된 기법을 앱 개발 툴로 구현하기에 적합하다. Fig. 6은 제안된 기법을 이용해 구현한 앱 개발 툴이다. ①번 영역은 기능과 센서 동작 모듈들이 존재한다. ②번 영역은 화면을 추가할 수 있는 화면 모듈들이다. ③번 영역은 편집기에 추가된 화면 모듈을 선택하고 다른 기능 및 센서 모듈을 추가하여 사용자가 앱에 보여질 화면을 만드는 부분이다. 이미지 모듈, 음원 재생 모듈, 타이머 모듈, 흔들기 동작을 인식하는 센서 모듈이 포함되어 있다. ④

Fig. 6. Implemented App Development Tools and the results of smart phone execution.

번 영역은 에디터 상에 추가되어 있는 화면 모듈들이 다른 모듈들을 통해 앱의 실행 화면이 전환되는 내용을 트리 형태로 표시되는 부분이다. ⑤번은 앱 개발 툴을 통해 제작한 앱을 Galaxy Nexus 디바이스에서 실행한 화면이다.

다음으로는 모듈이 사용되는 과정을 이미지, 음악 재생 기능 기능의 추가를 통해 확인한다. 그리고 센서 모듈을 사용하여 센서를 사용하는 방법을 설명하고 모듈과 화면 모듈의 연결 방법을 살펴볼 것이다. 마지막으로 모듈 연결을 통해 앱 처리과정이 트리구조로 표현되는 것을 설명한다.

Fig. 7은 사용자가 앱 화면에 이미지를 추가할 때 모듈의 사용 방법을 나타낸 화면이다. 화면 모듈(screen1)에 이미지 표시 모듈을 드래그한 드롭하면 이미지를 표시하는 기능이 화면 모듈에 추가된다

(①). 화면 모듈에 추가된 모듈들은 사용자의 정보 입력 작업을 필요로 할 경우 선택 정보 입력 부분을 통해 관련 내용이 표시된다. 사용자는 표시된 내용을 읽고 버튼을 누르면 모듈에 맞는 사용자 정보 입력 작업이 실행된다(③). 이때 실행되어 나타나는 화면은 사용자가 화면 모듈 내에 추가한 모듈의 내부에서 정의되어 있는 내용으로 3.2절에서 기술된 사용자 정보 입력 코드가 실행된다. 사용자가 선택한 정보는 모듈 내부 데이터로 저장한다. 사용자는 이미지 표시 모듈의 선택 정보로써 표시할 이미지 선택하는 과정을 마치면(④) 화면 모듈에 선택한 이미지가 표시된다(⑤).

Fig. 8은 화면 모듈 내에 음악 재생 기능을 추가할 때의 과정이다. 이 과정은 화면 모듈에 다른 기능 모듈이 이미 존재할 때의 상황을 나타낸다. 이미지

Fig. 7. Add an image print function on Screen Module.

Fig. 8. Add sound playback function on Screen Module.

표시 기능 추가 과정과 동일하게 사용자가 음원 재생 모듈을 선택하여 화면 모듈 내부로 드래그 하면 기능이 추가된다(①). 화면 모듈 내부에 모듈이 추가되면 사용자 정보 입력 부분에 표시되어 있던 내용은 음원 재생 모듈에서 필요한 사용자 정보 입력에 대한 내용으로 대체된다(②). 이전의 표시 내용은 “이미지 선택”과 관련한 내용이었으나 “음악 선택”에 대한 내용으로 변경된 것을 알 수 있다. 해당 버튼을 클릭하면 음원 선택에 대한 팝업 윈도우가 표시된다(③). 현재 작업 중인 화면 모듈에서 재생할 음원 모듈 선택 작업을 마치면(④) 음악 재생 기능을 추가하는 과정이 완료된다(⑤).

Fig. 9는 사용자가 스마트폰을 5회 흔들는 동작을 스마트폰 센서로 인식하는 기능을 화면 모듈에 추가

할 때의 과정을 나타낸 것이다. 사용자는 흔들기 센서 모듈을 작업 중인 화면 모듈에 추가하면(①) 화면 모듈의 우측 상단에 추가한 센서 모듈이 표시된다(②). 사용자가 선택한 모듈에서 필요한 선택 정보는 사용자 정보 입력 부분에 표시된다(③). 정보 입력 버튼을 누르면 추가한 센서 모듈에 대한 정보 입력 창이 활성화되며 사용자로부터 스마트폰을 몇 회 흔들었을 때 반응하도록 할 것인지를 입력 받게 된다(④). 정보 입력을 마치면 최종적으로 해당 동작이 인식 되었을 때 이동할 화면을 지정하도록 안내하게 된다(⑤).

Fig. 10은 사용자가 센서 모듈을 통해 지정한 동작이 인식 되었을 때 다른 화면 모듈로 이동하도록 설정하는 과정을 나타낸 것이다. 사용자는 쉬프트

Fig. 9. Add shake motion recognition function on Screen Module.

Fig. 10. Connect sensor module and Screen Module to be switched.

(Shift) 키를 이용하여 대상 모듈을 선택하고(①) 이동되어질 화면 모듈을 선택 한 후(②) 모듈 연결 버튼을 누르면(③) 모듈과 화면 모듈이 연결된다. Fig. 10의 우측의 화면은 (①,②,③)의 과정을 통해 연결된 모듈과 화면 모듈을 3.4절에서 기술한 실행 흐름 전환 관계에 따라 트리 구조로 보여주는 것이다. 이 과정은 사용자에게 모듈과 화면을 연결하는 즉시 갱신되어 표시된다.

Fig. 11은 사용자가 모듈(기능, 센서 모듈)과 화면 모듈의 연결을 통해 앱을 만들 때 사용자에게 앱이 실행되는 과정을 3.4절에서 기술한 표시 방법을 사용하여 보여주는 화면이다. 사용자가 작업 중인 화면 모듈은 크게 보이고 다른 화면 모듈은 상대적으로 작게 표시하며 모듈을 선택할 때 해당 모듈을 통해

이동할 화면을 화면상에 강조하여 보여주게 된다. 각 화면을 트리구조로 표시하면서 언제 화면이 전환되게 되는지에 대한 설명을 초록색으로 구분하여 표시한다. 이 과정은 사용자가 만들고 있는 앱이 실행되는 전체 흐름을 쉽게 이해하도록 한다.

4.2 결과 및 비교 분석

기존 앱 개발 방법과의 비교를 위해 응용 앱 개발 시나리오를 본 논문에서 제안한 앱 제작 기법을 구현한 앱 개발 툴과 App Inventor를 통해서 비교했다. App Inventor의 경우 밝기 센서를 지원하지 않았기 때문에 버튼으로 대체한 후 앱 제작 과정에서 필요한 작업을 비교하였다. Table 4는 기존 시스템인 App

Fig. 11. App display of the execution process.

Table 4. Comparison with the existing method

Item	App Inventor	Proposed System
Required components (total)	11	9
Resource assignment (total)	11	9
Detail setting (total)	12	None
Media Upload (total)	3	None
Project Source	Not supported	Support ADT conversion, Output Source

Table 5. Comparison of sensor usage

Item	App Inventor	Proposed System
Supported sensor type	3	5
Supported Form	Sensor component	sensor module with recognizable behavior defined
Method of sensor usage	Direct control of the sensor data value (Using the logic block)	Select action to recognize

Inventor와 비교한 내용이다. App Inventor의 경우 화면 구성과 배치된 구성 요소들에 대해서 표시할 리소스나 동작에 대한 논리를 지정하기 위해 화면 구성 이외에 부가적인 작업이 본 논문에서 구현한 시스템보다 더 많은 것을 알 수 있다. 또한 시스템의 동작이 원격 서버를 통해서 동작하게 됨으로써 앱 제작을 위해 화면에 표시할 데이터들을 서버에 업로드한 이후에 사용을 해야 했다. 전송 방법에는 작성한 앱을 다른 사용자에게 보내주고 싶다면 본 논문에서 제작한 시스템은 이메일을 통해서 원하는 사람에게 쉽게 전송할 수 있다. 하지만 App Inventor는 PC를 통해 안드로이드 패키지 파일을 내려 받은 후 파일 복사 등을 통해 이를 사용자에게 직접 공유하는 과정이 필요 했다. App Inventor의 경우 제작한 앱에 대한 프로젝트 정보 전체가 원격 서버에 저장되므로 이를 사용자가 소유할 수 없지만 본 논문의 구현은 사용자의 PC에서 실행되며 ADT(Android Development Tools)환경에 맞는 프로젝트 형식과 안드로이드 소스로 출력되므로 이후 직접 앱 개발이 가능하다.

Table 5는 각 시스템에 대해서 스마트폰에 부착된 센서를 사용할 때의 방법을 비교한 것이다. 제안된 시스템의 경우 총 5종의 센서를 사용할 수 있으며 사용자가 앱을 제작하는 과정에 추가할 동작을 센서를 통해 인식할 수 있는 사용자의 행동의 명칭을 단위로 구현된 모듈을 제공한다. 따라서 센서 모듈을 화면에 추가하는 것만으로 추가적인 센서 값의 처리 기술 없이 쉽게 센서를 사용할 수 있다. 하지만 App Inventor의 경우 직접 3종의 센서로부터 들어오는 센서 데이터를 블록 에디터를 사용해 직접 처리해 사용해야 한다. 따라서 비개발자가 앱 개발 과정에서 센서를 통해 특정한 동작을 인식하여 사용하려면 roll(수평 기울기), pitch(수직 기울기), azimuth(방위)값과 같은 센서 데이터 처리에 관련된 지식과 논리 블록을 사용한 제어를 필요로 하게 된다.

5. 결 론

비개발자인 사용자가 지금까지 앱을 개발하기 위해서는 프로그래밍을 익혀야 했다. 또한 비개발자를 위한 앱 개발 도구가 존재하지만 많은 설정 동작을 필요로 하고 센서에 대한 지원이 부족하였다.

본 논문에서는 기능 구현과 스마트폰의 센서를 사용하기 쉬운 앱 개발 시스템을 설계하고 구현하였다. 구현된 시스템은 기능 모듈과 센서 모듈들을 한 개의 화면으로 조합하면 구현하고자 하는 기능을 만들 수 있고 각 화면은 모듈과 화면간의 연결을 통하여 다른 화면으로 이동하도록 제어할 수 있다. 또한 센서 모듈은 스마트폰의 센서를 통해 인식되는 사용자의 행동을 단위로 구성되어 사용이 쉽고 추가적인 센서 제어 및 데이터 추출에 대한 기술이 필요 없다.

앱의 생명 주기, 즉 처리되는 순서는 사용자가 기능 모듈을 조합해 만든 각 화면들을 트리구조로 표시하고, 앱 실행 중 앱의 실행 흐름이 변경되는 시점을 기능 모듈의 이벤트 이름과 화살표를 사용해 이동되어지는 화면으로 연결시켜 보여주기 때문에 사용자는 앱이 실행되는 과정을 쉽게 파악할 수 있다. 구현된 툴을 이용해 제작된 앱은 이메일을 통해 다른 사람에게 전송할 수 있다. 또한 전송에 소요되는 시간이 짧기 때문에 사용자는 이벤트 및 기념일 앱과 같은 사용자 스토리 기반 앱을 제작하고 빠른 공유가 가능하다.

본 논문에서 구현한 모듈 방식을 이용한 앱 제작 시스템은 모듈, 개발, 구성 정보, 빌드의 단계가 구분되어 있기 때문에 다른 스마트폰 운영체제를 위한 앱 개발이나 비개발자를 위한 소프트웨어 개발 시스템 구현과 관련된 부분에 응용이 가능할 것이다. 또한 구현된 시스템은 앱 개발과 개인간 전송을 통한 앱 공유 환경을 지원하기 때문에 앱을 통한 소셜 플랫폼으로 활용될 수 있을 것이다.

REFERENCE

- [1] Ministry of Science, ICT and Future Planning, *Wired and Wireless Communication Service Subscriber Statistics*, 2013.
- [2] Korea Internet & Security Agency, *Survey on the Use of Smartphones Second Half of 2012*, 2013.
- [3] B.H. Kim, "Computer Programming Education using App Inventor for Android," *Journal of the Korea Institute of Information and Communication Engineering*, Vol. 17, No. 2, pp. 467-472, 2013.
- [4] D. Wolber, H. Abelson, E. Spertus, and L. Looney, *AppInventor: Create Your Own Android Apps*, O'Reilly, Calif., 2011.
- [5] Create App (I), <http://www.appcookkr.co.kr/AppInstance/Index?aii=18>, 2013.
- [6] App Technology and Vision in Age of Mobile, <http://www.wonderfulsoft.org/ups/wonderfulsoft2/images/mobileapp.pdf>, 2013.
- [7] J.S. Park and S.B. Cho, "The Effect of Teaching Scratch in Introductory Programming Course," *The Korea Society of Digital Policy and Management Journal of Digital Convergence*, Vol. 10, No. 9, pp. 449-456, 2012.
- [8] K.W. Park, N.H. Ryu, and E.K. Kim, "Language Education System with Structured Programming," *Journal of Korea Institute of Electronic Communication Sciences*, Vol. 5, No. 5, pp. 459-464, 2010.

- [9] H.K. Yang and H.S. Yong, "Physical Activity Recognition using Accelerometer of Smart Phone," *Proceeding of the Korean Institute of Information Scientists and Engineers Fall Conference, Mobile Application and System*, pp. 7-9, 2012.
- [10] J.I. Choi and H.S. Yong, "Activity Data Modeling and Visualization Method for Human Life Activity Recognition," *Journal of Korea Multimedia Society*, Vol. 15, No. 8, pp. 1059-1066, 2012.
- [11] Graphical Editing Framework, <http://www.eclipse.org/gef>, 2013.

박 주 연

2007년 3월~2012년 2월 조선대
학교 컴퓨터공학과 학사
2012년 3월~2014년 2월 전남대
학교 전자컴퓨터공학과
석사
관심분야: 소프트웨어 공학, 멀티
미디어 응용 시스템, 임베
디드 소프트웨어 등

박 성 모

1977년 서울대학교 전자공학과
학사
1979년 한국과학기술원 전기 및
전자공학과 석사
1988년 노스캐롤라이나 주립대
학 전기 및 컴퓨터공학과
공학박사
1979년~1984년 한국전자기술연구소 설계개발부 선임
연구원
1988년~1992년 올드도미니언대학교 전기 및 컴퓨터공
학과 조교수
1992년~현재 전남대학교 전자컴퓨터공학부 교수
관심분야: 멀티미디어 프로세서 구조, SoC 설계, 영상압
축, 임베디드 시스템 등