JKIICE Journal of the Korea Institute of Information and Communication Engineering

한국정보통신학회논문지 Vol. 22, No. 2: 221~226, Feb. 2018

# 앱인벤터 오픈소스 수정 · 구현

김병호*

# Revision and Implementation of App Inventor Open Source

**Byungho Kim***

Department of Computer Science, Kyungsung University, Busan 48434, Korea

## 요 약

오픈소스는 소스 코드를 볼 수 있고 수정할 수 있다는 점에서 기존 코드의 버그 수정이나 새로운 아이디어의 추가가 가능하다. 오픈소스는 이를 이용한 상업용 소프트웨어 개발은 물론 교육용으로도 그 활용 범위가 매우 넓다. 교육용 블록 프로그래밍 환경의 하나인 앱인벤터는 앱인벤터 오픈소스를 통해 자체 로컬 서버의 구축은 물론 플랫폼 자체도 수정할 수 있다. 본 논문에서는 앱인벤터 오픈소스를 로컬 서버에 구현한다. 특히 교육기관의 특성을 반영한 로컬 서버 구축을 위해 앱인벤터 오픈소스 플랫폼을 수정하고 그 과정에 대하여 기술한다. 이를 통해 학생들에게 보다 친숙한 교육 환경을 제공할 수 있고 나아가 학생들의 요구를 반영한 나만의 프로그래밍 블록 추가와 같은 능동적인 대처를 통해 교육 만족도도 높일 수 있다.

## ABSTRACT

Open source software allows revising and adding a new idea on them as well as debugging themselves. The open source software can be utilized as various educational programs as well as development of a commercial software products using them. MIT App Inventor, one of the block-based visual programming environments for education, supports building a local server and even revision of the platform itself by providing its own source codes. In this paper we implement App Inventor open sources on the local server. Especially we present how to revise and implement a new App Inventor platform reflecting their own requirements in schools or educational institutes and how to build a local server operating the revised App Inventor platform. They can provide more interactive educational environments to students and improve educational satisfaction through the active customizing of the platform itself such as adding new programming block components requested by students.

**키워드** : 오픈 소스, 앱 인벤터, 프로그래밍 블록 요소 추가, 로컬 서버, OER

Keyword : Open Sources, App Inventor, Revising of Programming Block Components, Local Build Server, OER

**Received** 17 December 2017, **Revised** 3 January 2018, **Accepted** 29 January 2018

* **Corresponding Author** Byungho Kim(E-mail:bkim@ksu.ac.kr, Tel:+82-51-663-4787)
Department of Computer Science, Kyungsung University, Busan 48434, Korea

Open Access http://doi.org/10.6109/jkiice.2018.22.2.221 pISSN:2234-4772

ⓒThis is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License(http://creativecommons.org/li-censes/by-nc/3.0/) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.
Copyright Ⓒ The Korea Institute of Information and Communication Engineering.

# Ⅰ. 서 론

오픈소스는 소스 코드가 공개된 프로그램이다[1]. 프리웨어가 무료로 사용할 수 있는 프로그램인 반면 오픈소스는 소스 코드를 볼 수 있고 수정할 수 있다는 점에서 기존 코드의 버그 수정이나 새로운 아이디어의 추가가 가능하다. 따라서 오픈소스는 이를 이용한 상업용 소프트웨어 개발은 물론 교육용으로도 그 활용 범위가 매우 넓다[2].

한편 2018년부터 초·중등 정규과정에 소프트웨어 교육이 의무화된다[3]. 컴퓨터 프로그래밍을 처음 접하는 초·중등 학생들에게 컴퓨팅 사고는 물론 흥미를 유발하는 효율적인 코딩 교육은 쉬운 일이 아니다[4]. 주로 언급되는 교육용 언어로는 스크래치[5], 엘리스[6], 엔트리[7], 앱인벤터[8]가 있다. 이들은 모두 블록 기반 프로그래밍 언어로써 영어를 사용하는 C나 자바와 같은 범용 프로그래밍 언어와 달리 언어에 대한 장벽 없이 프로그래밍 개념 자체에 보다 집중할 수 있다는 장점이 있다.

초·중등 소프트웨어 교육을 위한 프로그래밍 언어를 선택할 때 교육 제공자 입장에서 고려해야 할 중요한 이슈 중 하나는 학생들이 프로그램을 개발하고 실행하는 프로그램 개발 환경의 편의성이다. 위에서 언급한 4개 언어 모두 웹 기반 클라우드 환경이고 한글도 지원하지만 개발 환경 자체의 변경, 예를 들어 코딩 실습 플랫폼 자체를 학교 전용으로 사용한다거나 그 웹페이지에 학교명 명시, 또는 교내 학생들만을 위한 사설 작품 갤러리 개설 등을 위한 플랫폼 수정은 대부분 불가능하다.

반면에 앱인벤터는 MIT 대학에서 공개 플랫폼도 운영하고 있지만[9] 앱인벤터 오픈소스를 통해 자체 로컬 서버의 구축은 물론 플랫폼 자체도 수정할 수 있다. 예를 들어 기존 MIT 앱인벤터는 구글 클라우드 환경에서 수행되기 때문에 구글 로그인이 필요한데 14세 미만은 계정 생성이 쉽지 않다. 이는 자체 로컬 서버 구축을 통해 구글 계정 없이도 사용하게 할 수 있다[10].

본 논문에서는 앱인벤터 오픈소스를 이용하여 단순히 로컬 서버의 구축뿐만 아니라 교육기관의 특성을 반영한 프로그래밍 플랫폼 자체의 수정 방법과 과정에 대하여 기술한다. 이를 통해 학생들에게 보다 친숙한 교육 환경 제공은 물론 나아가 학생들의 요구를 반영한 나만의 프로그래밍 블록 추가와 같은 능동적인 대처를 통해 교육효과를 극대화할 수 있다.

본 논문의 구성은 다음과 같다. 2장에서 주요 블록 프로그래밍 언어들에 대하여 살펴보고, 3장에서 앱인벤터 오픈소스를 이용하여 새로운 프로그래밍 블록 요소를 추가하는 과정과 로컬 서버 구현에 대하여 설명하고, 4장에서 구현 결과와 향후 활용방안을 설명하고 마친다.

# Ⅱ. 관련 연구

## 2.1. 스크래치

스크래치는 이미지나 사운드를 비주얼 프로그래밍 블록들을 이용해 제어할 수 있게 함으로써 컴퓨터에 대한 지식이 전혀 없는 일반인이나 어린이들도 컴퓨터 프로그래밍의 개념을 이해할 수 있도록 도와주는 교육용 프로그래밍 언어이다[5]. 2007년 MIT 미디어랩의 Lifelong Kindergarten 그룹에서 개발하였으며[11], 비주얼 프로그램 개발환경과 커뮤니티 기반 웹 인터페이스로 구성되어 있다.

스크래치 시스템의 가장 큰 장점은 온라인 커뮤니티이다. “상상, 프로그래밍, 공유”라는 표어 아래 스크래치를 사용하는 이용자들이 정보와 개발 프로젝트를 공유하는 곳이다. 2007년 스크래치 웹사이트 개설 이후 회원수는 80여만 명, 공개된 프로젝트는 180여만 개에 달하며 회원들의 연령대도 11세에서 50세에 걸쳐 있고 평균 연령은 15세이다[12].

## 2.2. 엘리스

엘리스는 스크래치처럼 컴퓨터 프로그래밍에 관한 지식이 없어도 애니메이션으로 이야기를 구성하거나 대화형 게임 등을 만들 수 있는 3D 비주얼 프로그래밍 환경이다[6]. 각각의 블록들은 자바, C++, C# 등 객체지향 프로그래밍 언어의 표준 문장을 따르고 있어 객체지향 프로그래밍 언어 개념을 학습하는데 유용하다.

## 2.3. 앱인벤터

앱인벤터는 안드로이드 스마트폰 앱을 개발할 수 있는 비주얼 프로그래밍 도구로써 기본 플랫폼은 MIT 미디어랩에서 개발한 스크래치에 기반하고 있다[8]. 초기에는 구글에서 운영하였으나 2012년부터 MIT로 이전되었다. 일반적인 안드로이드 SDK로 개발한 앱과 마찬

가지로 앱인벤터로 만든 앱도 안드로이드 스마트폰 설치는 물론 앱스토어 등록도 가능하다[9].

앱인벤터는 앱 설계기, 블록 편집기, 에뮬레이터의 3개 프로그램으로 구성된다.

2.3.1. 앱 설계기

앱 설계기는 버튼, 이미지, 소리, 동영상 등 앱에서 사용할 요소들을 배치하고 설계하는 도구로써 별도의 프로그램이 아닌 앱인벤터 홈페이지에서 실행한다. 일반 안드로이드 SDK에서의 레이아웃에 해당된다.

앱 설계기는 그림 1과 같이 세로로 구분된 4개 영역, 즉 팔레트, 뷰어, 콤포넌트, 속성 영역으로 구성된다.

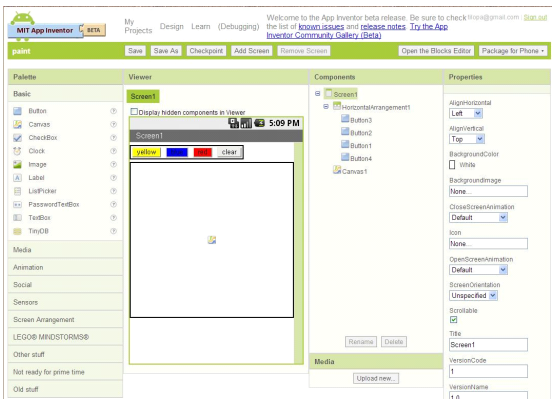

**Fig. 1** Snapshot of App Inventor designer

2.3.2. 블록 편집기

블록 편집기는 비주얼 코드 블록들을 사용하여 앱 요소들의 동작을 작성하는 도구로써 일반 안드로이드 SDK에서의 자바 프로그래밍에 해당된다.

스마트폰 앱의 동작 방식은 이벤트 기반이다. 주변 상황의 어떤 변화, 즉 발생 가능한 각각의 이벤트에 원하는 동작을 연결하는 방식인데 이 작업은 블록 편집기에서 이루어진다.

## Ⅲ. 앱인벤터 오픈소스 수정 · 구현

본 논문에서는 앱인벤터에 새로운 콤포넌트를 추가한다. 이를 위해 앱인벤터 오픈소스를 수정하고 다시 컴파일 해야 한다. 구현 결과를 먼저 보면 그림 2와 같다. 기존 MIT 앱인벤터 앱 설계기 화면인 그림 1과 비교하면 그림 2에서는 원으로 표시한 영역에 새로운 콤포넌트들이 추가된 것을 볼 수 있다.

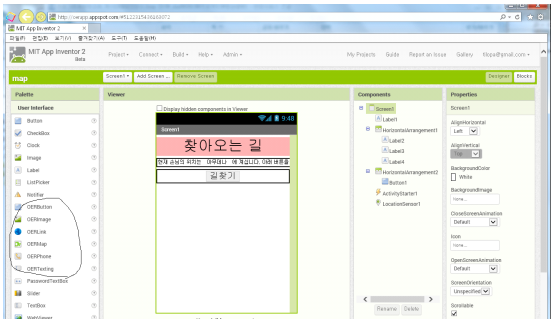

**Fig. 2** A New App Inventor with Newly Added Components Marked by a Circle

본 논문에서 개발한 앱인벤터는 한국방송통신대학교 OER의 “나만의 모바일앱 만들기” 과정의 실습환경으로 활용중이며 이 과정은 컴퓨터 프로그래밍 경험이 없는 일반인 대상이다[13]. 과정 홈페이지는 그림 3과 같다.

**Fig. 3** KNOU OER Homepage for "Building my own mobile app" Contents

본 장에서는 앱인벤터 오픈소스를 수정하여 새로운 콤포넌트를 추가하는 과정을 기술한다. 1절과 2절에서 앱인벤터 오픈소스의 구조와 구성을 파악하고, 3절에서 기존 콤포넌트들은 어떻게 구성되어 있는지 살펴본다. 다음으로 수정한 소스를 컴파일하기 위한 환경구축 과정에 대해 살펴보고 5절과 6절에서 추가한 콤포넌트들

과 앱인벤터 오픈소스 수정에 대해 살펴본다.

### 3.1. 앱인벤터 오픈소스 구조

앱인벤터 오픈소스는 그림 4와 같이 여러 소프트웨어 플랫폼들로 구성되어 있다[14].

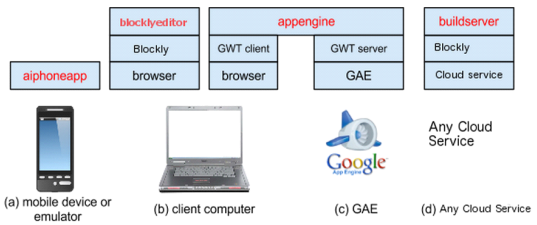


**Fig. 4** Platform Architecture of the App Inventor Open Source

새로운 콤포넌트를 추가하기 위해 수정해야 하는 플랫폼은 그림 4에서 빨간색 글씨로 표시된 appengine, buildserver, blocklyeditor, aiphoneapp이다. appengine 플랫폼은 앱인벤터 앱 설계기와 블록편집기 실행을, buildserver 플랫폼은 컴파일과 패키징을 담당한다.

### 3.2. 앱인벤터 오픈소스 구성

앱인벤터 소스코드의 주요 디렉토리 구조는 아래와 같다[15].

- appengine: 앱인벤터가 클라이언트 브라우저에서 동작하도록 지원하는 구글 웹 기반 플랫폼. 클라이언트와 서버 간의 통신과 프로젝트의 저장, 관리, 패키징, 배포를 위한 외부 buildserver간의 통신을 담당
- blocklyeditor: 앱인벤터 블록편집기
- buildserver: 안드로이드 앱 패키징을 담당
- components: 앱인벤터 콤포넌트들의 표현, 기능, 구현, 스크립트 등에 관한 모든 클래스 포함
- aiphoneapp: 안드로이드 스마트폰 또는 PC 에뮬레이터에서 수행 지원
- aiplayapp: 안드로이드 스마트폰 에뮬레이터
- common: 프로그램 전반의 공통 상수 및 클래스들을 포함
- docs: 프로그램 설명
- lib: 외부 라이브러리

### 3.3. 앱인벤터 콤포넌트

콤포넌트는 앱에서 사용되는 다양한 요소들을 모듈로 구현한 것이다. 버튼, 레이블, 캔버스 등 80여개의 콤포넌트가 있다[15]. 각 콤포넌트를 규정하는 속성들을 보면 표 1과 같다.

**Table. 1** Properties of a Component

| Properties | Descriptions | Sample Values |
|---|---|---|
| name | component name | Accelerometer Sensor |
| version | increase as update | 1 |
| category String | categories | SENSORS |
| help String | description string | |
| non Visible | nonvisible or not | true |
| icon Name | icon image file name | images/a.png |

### 3.4. 소스코드 개발환경

수정한 앱인벤터 소스코드 컴파일에 필요한 개발환경은 아래와 같다. 앱인벤터 오픈소스는 자바 언어로 작성되어 있고 컴파일 환경은 리눅스이다.

1) 운영체제: Fedora 19, 64비트 CPU
2) 자바 JDK: Oracle JDK. wget으로 jdk.tar.gz 설치, /etc/profile에 path 설정
3) ant: wget으로 tar.gz 설치, /etc/profile에 path 설정
4) github: 앱인벤터 소스코드 설치 용도
5) 자바 앱 엔진 SDK: 자바 버전 구글 앱 엔진을 /usr /local에 설치
6) 파이선 설치
7) 일부 앱인벤터 소스는 32비트 라이브러리를 사용하므로 yum으로 해당 라이브러리 추가 설치
8) 32비트 컴파일 오류 발생 시 해당 라이브러리의 .686 버전으로 대체

### 3.5. 추가 콤포넌트들

본 논문에서 새로 추가한 6개 콤포넌트들의 명칭과 설명은 표 2와 같다.

**Table. 2** Descriptions of new Components

| Components | Descriptions |
|---|---|
| OERLink | direct web access with only URL |
| OERMap | navigation service from current location to desired spot |
| OERButton | OER-specialized button |
| OERImage | list of newly added images |
| OERPhone | direct phone call from Phonelist |
| OERTexting | direct texting from Phonelist |

추가된 콤포넌트들의 원래 콤포넌트는 표 3과 같다.

**Table. 3** The new Components and the Corresponding Original Components

| New Components | Original Components | |
|---|---|---|
| | Name | Palette Groups |
| OERLink | Web | Connectivity |
| OERMap | Map | Maps |
| OERButton | Button | User Interface |
| OERImage | Image | User Interface |
| OERPhone | PhoneCall | Social |
| OERTexting | Texting | Social |

OERLink, OERButton, OERImage, OERPhone, OERTexting은 각각 원래의 Web, Button, Image, PhoneCall, Texting 콤포넌트의 기능과 동일하지만 명칭과 아이콘 이미지를 바꾼 것이다. 예를 들어 원래의 Web 콤포넌트는 안드로이드 SDK에서 WebView 객체에 해당하는 것으로써 새로 열고자 하는 웹페이지의 URL만 지정하면 해당 웹페이지를 여는 콤포넌트이다. 본 논문에서는 OERLink라는 콤포넌트로 추가하였다. OERMap은 원래의 Map 콤포넌트를 확장한 것으로써 목적지만 지정하면 현재 위치에서 목적지로의 경로를 보여주는 콤포넌트이다. 이 콤포넌트는 특히 본 논문의 결과를 적용할 방송통신대학 OER 과정 수강생들 중에 은퇴 후 자기 사업을 준비하는 소상공인들이 많다는 점에 착안하여 자신의 가게 주소를 입력하면 출발지별로 찾아오는 길 안내 지도를 자동으로 생성하는 기능의 콤포넌트이다.

### 3.6. 앱인벤터 소스코드 수정

표 2의 콤포넌트들을 추가하기 위해 수정해야할 소스코드, 즉 자바 소프로그램들과 수정 순서는 아래와 같다[16-20].

1) appinventor/appengine/src/com/google/appinventor/images 폴더에 추가 콤포넌트들의 이미지 파일 추가
2) appinventor/appengine/src/com/google/appinventor/client/Images.java에 6개 이미지 처리를 위한 메소드 추가
3) 추가 콤포넌트에 대한 클래스 코드 작성: /usr/local/appinventor/appinventor-sources/appinventor/components/src/com/google/appinventor/components/runtime 폴더에서 OERButton.java, OERWeb.java, OERMap.java, OERPhone.java, OERImage.java, OERTexting.java 코드 작성
4) /usr/local/appinventor/appinventor-sources/appinventor/components/src/com/google/appinventor/components/common/YaVersion.java 코드 수정
5) /usr/local/appinventor/appinventor-sources/appinventor/appengine/src/com/google/appinventor/client/editor/simple/components에 클래스 코드 추가: MockOERButton.java, MockOERImage.java, MockOERMap.java, MockOERPhone.java, MockOERTexting.java, MockOERWeb.java
6) 위 각각에 대해 클래스 이름, 생성자 이름, super() 호출 부분에 콤포넌트 아이콘 이미지 추가
7) /usr/local/appinventor/appinventor-sources/appinventor/appengine/src/com/google/appinventor/client/editor/simple/palette/SimpleComponentDescriptor.java 코드 수정
7) /usr/local/appinventor/appinventor-sources/appinventor/appengine/war/WEB-INF/appengine-web.xml에 설정 수정
8) buildserver 생성
9) YaVersion.java 수정
10) YOUNG_ADNNDROID_VERSION 버전을 높이고, 추가된 6개 콤포넌트 버전을 2로 설정
11) /usr/local/appinventor/appinventor-sources/appinventor/blockslib/src/openblocks/yacodeblocks/BlockSaveFile.java 코드 수정
12) /ai/appengine/war/Ya-tos.html에서 “Terms of service” 사용하지 않도록 변경

## Ⅳ. 구현 결과 및 활용 방안

수정·구현한 앱인벤터는 oerapp.appspot.com에 설치하였다[21]. 결과는 그림 2와 같다. 그림에서 원으로 표시한 부분이 새로 추가된 6개의 콤포넌트들이다.

본 논문에서는 MIT 앱인벤터에 새로운 콤포넌트를 추가한 자체 앱인벤터 시스템을 구축하였다. 이를 위해 앱인벤터 오픈소스를 수정·개발하고 자체 로컬 서버에 설치하였다. 새로 구축된 앱인벤터는 한국방송통신대학교 OER의 "나만의 모바일앱 만들기" 과정의 앱 개발 실습환경으로 적용되었다.

앱인벤터와 같은 블록 프로그래밍 플랫폼에서 새로운 블록 요소의 추가는 학교나 기관 고유의 특성을 반영할 수 있어 학생들에게 보다 친숙한 교육 환경을 제공할 수 있다. 또한 학생들의 요구에 따라 프로그래밍 교육 플랫폼 자체를 수정하는 맞춤형 교육을 통해 교육 만족도를 높일 수 있다.

## References


[1] Wikipedia, Open-source Software [Internet]. Available: https://en.wikipedia.org/wiki/Open-source_software.

[2] M. Lee, "Open Source Software Education," *Communications of the Korean Institute of Information Scientists and Engineers,* vol. 35, no. 9, pp. 17-25, Sep. 2017.

[3] Digital Daily, "Obligation of the Software Education for Elementary and Middle School by 2018," [Internet]. Available: http://www.ddaily.co.kr/news/article.html?no=132665.

[4] B. Kim, "Computer Programming Education using App Inventor for Android," *Journal of Korea Institute of Information and Communication Engineering*, vol. 17, no. 2, pp. 467-472, Feb. 2013.

[5] N. Resnick, et. al., "Scratch: programming for all," *Communications of the ACM*, vol. 52, no. 11, pp. 60-67, 2009.

[6] S. Cooper, W. Dann, R. Pausch, "Teaching Objects-first in Introductory Computer Science," *34th SIGCSE symposium on Computer science education*, pp. 191-195, 2003.

[7] Entry [Internet]. Available: https://playentry.org.

[8] D. Wolber, "App inventor and real-world motivation," *SIGCSE'11 - 42nd ACM Technical Symposium on Computer Science Education*, pp. 601-606, 2011.

[9] App Inventor [Internet]. Available: http://appinventor.mit.edu.

[10] S. Lee, K. Seong, "Constructions of Local App Inventor 2 Private Server for Android App Development," *Symposium of the Korean Society of Computer and Information*, vol. 22, no. 2, pp. 61-64, 2014.

[11] Scratch imagine program share [Internet]. Available: http://info.scratch.mit.edu/Scratch_Credits.

[12] MIT Media Lab, Press Announcement: Scratch [Internet]. Available: http://www.media.mit.edu/press/scratch.

[13] Korea National Open University OER [Internet]. Available: http://oer.knou.ac.kr.

[14] MIT App Inventor Sources [Internet]. Available: http://appinventor.mit.edu/appinventor-sources.

[15] E. Spertus, J. Dominguez, "App Inventor Developer Overview" [Internet]. Available: https://docs.google.com/document/d/1hIvAtbNx-eiIJcTA2LLPQOawctiGIpnnt0AvfgnKBok/pub.

[16] App Inventor Forum [Internet]. Available: https://groups.google.com/forum/#!categories/mitappinventortest/another-app-inventor-service.

[17] App Inventor Open Source Development [Internet]. Available:https://groups.google.com/forum/#!forum/app-inventor-open-source-dev.

[18] AI Live Complete [Internet]. Available: http://sourceforge.net/projects/ailivecomplete.

[19] App Inventor Server [Internet]. Available: http://sites.jsoft.com/appinventor/home/gsuac/server#TOC-server-options.

[20] H. Lee, J. Kim, "A Study on the Effects of the Agile Methods Practices on the Success of Open Source Community," *Asia-pacific Journal of Multimedia Services Convergent with Art, Humanities, and Sociology*, vol. 7, no.4, pp. 217-224, April 2017.

[21] OER App Inventor [Internet]. Available: http://oerapp.appspot.com.

김병호(Byungho Kim)

1990년: 연세대학교 전산과학과 학사
1992년: KAIST 전산학과 석사
1997년: KAIST 전산학과 박사
2007년-현재: 경성대학교 소프트웨어학과 부교수
※관심분야 : 컴퓨터구조, 모바일 컴퓨팅, 인공지능