

iPhone と Android アプリの共通開発支援ツール

村山 優弥[†] 佐藤 輝久[†] 瀬黒 雄作[†] 平野 竜希[†]

平松 貴宏[†] 三澤 拓也[†] 紫合 治[†]

東京電機大学 情報環境学科[‡]

1. はじめに

近年、スマートフォンの普及により、iPhone、Android アプリケーションの需要が高まっている。また、iPhone や Android のアプリケーション開発環境の構築のしやすさにより、今やアプリケーションも自分で開発を行う時代となっている。[1]しかし、開発言語や開発環境などの違いから、双方のスマートフォンで使えるアプリケーションを作成するためには、それぞれのアプリケーション開発の方法を学ばなければならず、開発初心者には敷居が高いと思われる。

そこで、本研究では、プログラミングを学んでいないアプリ開発初心者でも作成できる GUI(Graphical User Interface)を作成し、開発したアプリを Android、iPhone 双方に変換できるシステム開発支援ツールの開発を行った。

2. iPhone と Android の違い

iPhone と Android の開発環境の違いとして、iPhone 側は GUI を主に用いて視覚的に作成することができ、アプリ開発初心者には作成しやすい点がある。しかし、Objective-C という言語は認知度としては低く、多少なりプログラミング言語の学習が必要である。[2].Android 側でも GUI を用いて作成できるが、携帯電話の機種の違いにより画面サイズがそれぞれ違うため、ボタンやラベルなどの自由配置での設置がうまくいかない機種がある。そのため、開発者は様々な機種でも見られるよう、考慮してプログラミングを書かなければならない。[3].

3. システム構成

3.1 システムの概要

本システムの機能は、機種ごとの解像度、画面サイズの違いによる弊害を考慮して、GUI によるグリッド線を基準としたウィジェット設置機能、予め作成したテンプレートファイルに GUI データを代入し、それぞれのプログラムへと変換する機能で構築されている。

3.2 GUI について

本システムの GUI に含まれる機能として、ウィジェットの配置、ウィジェットデータの詳細変更、GUI データの保存と読み込み、変数登録、処理登録、Android プログラム、iPhone プログラムへの変換、以上のものがあげられる。これらの機能を用いて作成した電卓アプリを図 1 で示す。

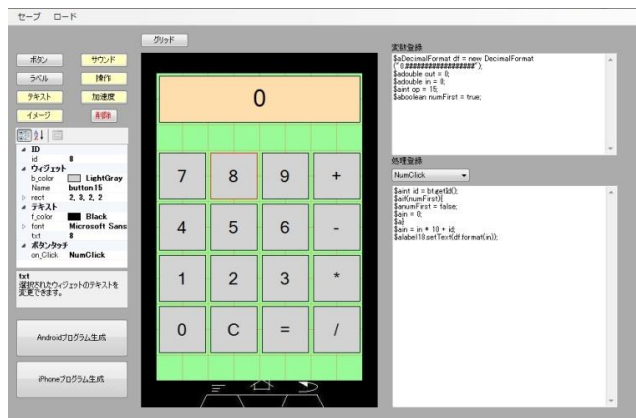


図 1. 開発支援ツールの GUI 画面 (電卓アプリ)

3.3 変換方式

本システムでの、プログラム変換方式の構成を図 2 で示す。まず GUI を用いて、画面に表示されているグリッド線上にあわせてウィジェットの配置を行い、そのデータを XML ファイルにて保存、読み込みを行う。

iPhone and Android apps common development tools

[†]Yuya Murayama, Teruhisa Sato, Yusaku Seguro, Ryuki Hirano, Takahiro Hiramatu, Takuya Misawa and Osamu Shigo

[‡]School of Information Environment, Tokyo Denki University

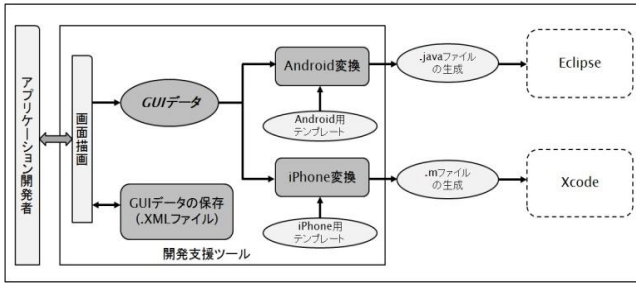


図 2. 変換方式の構成

次に、GUI データを基に、それぞれに対応したプログラム構成順にテンプレートファイルを読み込み、「\$0\$, \$1\$...」などの値の中に GUI 上のプロパティの値を代入する。(図 3、図 4) そして、Android 変換の場合は java ファイルを生成し、iPhone 変換の場合も同じようにテンプレートファイルを読み込み、m ファイルを生成する。Android と iPhone のプログラム構成の順番と変換の流れについて図 5 で示す。

```
Button $0$=new Button(this);
$0$.setId($1$);
$0$.setText("$2$");
$0$.setBackgroundColor(Color.$3$);
$0$.setTextSize($4$);
$0$.setGravity(Gravity.CENTER*$gravity$*);
layout.addView($0$,new AbsoluteLayout.LayoutParams
(gw*$5$-2*dgw, gh*$6$-2*dgh, gw*$7$+dgw, gh*$8$+dgh));
$0$.setOnClickListener(new $9$());
```

図 3. Android 用 button テンプレート

```
UIButton* $0$ = [UIButton buttonWithType:UIButtonTypeRoundedRect];
$0$.frame = CGRectMake(gw*$1$+dgw, gh*$2$+dgh, gw*$3$+dgw, gh*$4$+dgh);
$0$.setTitle:@"$5$" forState:UIControlStateNormal;
$0$.setTitleColor:[UIColor colorWithRed:$6$/255.0 green:$7$/255.0 blue:$8$/255.0 alpha:1.0f] forState:UIControlStateNormal;
$0$.titleLabel.setFont:[UIFont systemFontOfSize:$9$];
$0$.addTarget:self action:@selector($10$) forControlEvents:UIControlEventTouchUpInside;
$0$.tag = $11$;
[self.view addSubview:$0$];
```

図 4. iPhone 用 button テンプレート

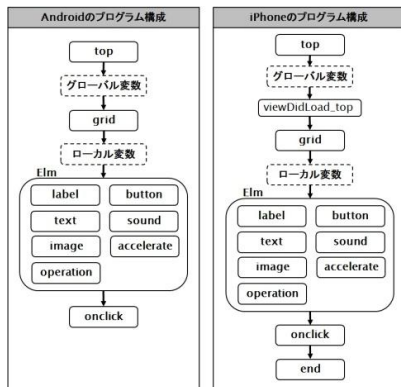


図 5. Android と iPhone の変換の流れ

4. 評価実験および調査報告

プログラミングが苦手な生徒に協力してもらい、電卓アプリを Xcode で作成した場合、Eclipse で作成した場合、本システムで作成した場合と比較し、作成にかかった時間を表 1 に示す。

表 1. 評価実験結果

	Xcode (iPhone)	Eclipse (Android)	本システム (iPhone+Android)
時間	1時間 18分	2時間 24分	1時間 2分

時間に差がでた理由は、Xcode では、ボタン、テキストとコードを接続する必要があるのに対して、本システムではその必要がなく関連付けることができるからであり、Eclipse では、電卓アプリに必要なプログラミングを調査するのにかかった時間が多かったためである。また、Android、iPhone のアプリを同時に作れるため、実際には作業時間は半分で済むことがわかる。

5. おわりに

本研究では iPhone, Android とともに同等のアプリの開発ができることを確かめられた。また、実験結果からアプリ開発初心者にも使いやすいシステムであることも確かめられた。

今回は、電卓アプリという簡単なアプリで実験を行なっているが、今後は高度なアプリ開発と比較、実験を行い、そこであげられた機能の追加、操作性の向上が課題としてあげられる。

[参考文献]

[1] インプレス R&D
<http://www.impressrd.jp/news/121120/kwp2013>
 [2] 所 友太, “iPhone プログラミング UIKit 詳解リファレンス”, 京セラコミュニケーションシステム, 2010.
 [3] 布留川 英一, “Android プログラミングバイブル”, ソシム, 2011.