

iOS アプリケーションのインターフェース開発を

支援する Web アプリケーション

WEB based application supporting the interface development of iOS application

田中貴之*1
Takayuki Tanaka

井上聡*1*2
Satoru Inoue

*1 埼玉工業大学大学院
Graduate School of Engineering, Saitama Institute of Technology

*2 埼玉工業大学
Saitama Institute of Technology

During the development of GUI based application, the more UI parts which are in the application, the more complex source code becomes. This causes a reduction in development efficiency. To support the programmers which engage in development of application, we purposed the WEB based application which generates the parts of source code by simple GUI action.

1. はじめに

スマートフォンなどで利用する GUI アプリケーションは複数の UI パーツから構成されており、そのアプリケーションの開発には、それぞれの UI パーツに相当したコードを記述する必要がある。言い方をかえれば、利用される UI パーツの数に比例して、そのソースコードは長文化することを意味し、多機能なアプリケーションになるほど、その開発にかかる時間や労力は増大していくことになる。また UI パーツはサイズや、表示する座標、色などさまざまな属性を持っており、アプリケーション上に単純にボタンを1つ配置するだけでも、複雑なソースコードの記述が必要になる(図1)。またユーザが利用しやすいインターフェースを持つアプリケーションを開発するためには、アプリケーション自体のデザインや、UI パーツを配置する位置、配置サイズなどは、試行錯誤での微調整が必要となり、そのたびに該当するパーツがもつ属性のパラメータをソースコード内で変更し、またテストを行うという反復作業は開発者の負担でもあり、開発の非効率化にもつながる問題点でもある。

2. 研究目的

スマートフォンの普及にともない、スマートフォン上で動作するアプリケーションの市場は拡大の一途をたどっている。また幅広い利用者のニーズに応えるべく、アプリケーションのカテゴリも多岐にわたっており、その開発スピードの高速化が求められているのは疑いのない事実である。しかしながら、アプリケーション開発の過程において、先述したような労力や開発期間を増大させる要因が存在しているため、生産性を高める手段を考慮する必要があるといえる。そこで本研究では、iOS アプリケーションのインターフェース部分に特化した、開発支援システムの構築を行う。GUI アプリケーションの開発では不可欠なインターフェースのデザインを、UI パーツのドラッグ&ドロップという直感的操作と開発者が UI パーツに加えたい属性の値を入力することにより、相当するソースコードを自動で生成できるような支援アプリケーションの開発を行う。このアプリケーションにより開発者の負担を軽減できると考える。また、本研究で提案す

るシステムは JavaScript で記述した WEB アプリケーションとして運用することにより、開発する際のプラットフォームを選ばず、また支援アプリケーションのインストールが不要で、常に最新のバージョンが利用可能という、運用コストを低減させる効果も期待される。そして iOS アプリケーションの開発経験が少ない人への支援ツールとしての役割も期待される



図1. ソースコードと実行結果

3. 研究概要

3.1 システムの流れ

本研究で提案するシステムは以下の通りである。

- (1) ソースコードを出力しようとする UI パーツを選択する。
- (2) 選択された UI パーツをドラッグし仮想の iOS 端末画面にドロップすると UI パーツが配置される。配置された UI パーツはドラッグ&ドロップすることにより再配置可能である。
- (3) ドロップした場所から仮想画面における座標情報を取得する。
- (4) UI パーツの配置に必要な関数名や大きさ、場所などを決める値を入力する。
- (5) 実行ボタンを押すことにより UI パーツ配置に必要なソースコードが出力される。拡大ボタンを押すことにより大きい文字、ウィンドウで表示が可能である。

同様に(1)から操作することにより複数の UI パーツのソースコードの出力が可能になる。

出力されたコードはコピーと書かれたボタンを押すことによってクリップボードにコピーされる。コピーされたコードを Xcode にペーストする。

3.2 UI パーツの配置について

UI パーツの名前が書かれた画像一覧からドラッグし仮想の iOS 画面にドロップする(図 2)。Web アプリケーション全体でドロップした場所の座標値を取得する。その値と仮想の iOS 画面の配置されている座標値を使い仮想の iOS 画面に UI パーツを配置する。配置すると同時にパラメータに座標値が入力される。入力される値は CGRect で配置を採用しているため UI パーツの左上の座標値である。配置する UI パーツの種類はドラッグ元の画像から自動的に選ばれる。UI パーツは再配置可能である。配置されている UI パーツをドラッグしドロップする。ドロップされた位置座標を取得し、再配置を行う。

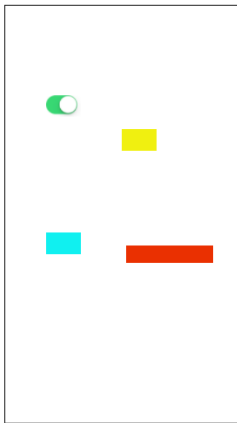


図 2. 仮想の iOS 画面

3.3 値の入力について

UI パーツの配置後、パラメーター一覧にサイズや関数の名前などの値を入力する必要がある(図 3)。

UI パーツが iOS の仮想画面に配置されるとパラメーター一覧は UI パーツに合わせた物に変更される。すでに配置されていたパラメーター一覧は非表示にされる。座標値以外の大きさや関数名などの値はユーザが入力する。

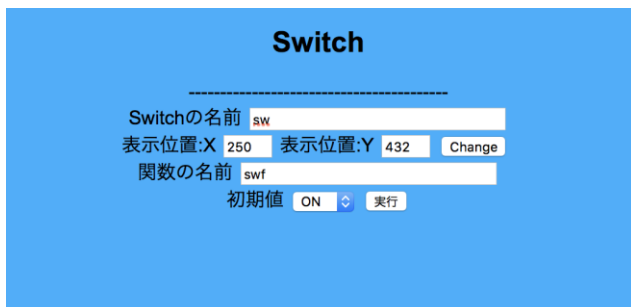


図 3. 入力が必要なパラメーター一覧

4. まとめ

数は限られてはいるが一部の UI パーツは容易にソースコードの出力が可能になった。図 5 の左の様にパーツを配置しパラ

メータに値を入力しコード出力を行った。そのコードを Xcode にペーストする(図 4)。Xcode にてプロジェクトをビルドしアプリケーションの実行を行うと入力した通りの結果が得られた(図 5)。しかし、開発現場で使われるには UI パーツの出力が少ないと思われる。出力される UI パーツのソースコードや入力可能な値を増やすことにより開発の幅を広げる必要があると考えられる。仮想画面での UI パーツの区別が色でしかできていない、一目ではわかりづらい。形の変更や文字を加えるなどすることにより、わかりやすく開発できると思われる。また、機能を追加することにより動作が重くなる場合やスムーズさを求める場合には、ネイティブアプリケーションとしての開発も必要だと思われる。それらにより開発に掛かるコストと時間をより抑えることができると考えられる。より多くの開発者に利用してもらうためには Safari、Google Chrome 以外の Web ブラウザへの対応、ユーザ管理またはキャッシュを使ったソースコードと仮想の iOS 端末画面の保存も必要と思われる。

```
import UIKit

class ViewController: UIViewController {

    override func viewDidLoad() {
        super.viewDidLoad()

        let btn = UIButton()
        btn.frame = CGRect(x: 101, y: 100, width:120, height:50)
        btn.backgroundColor = UIColor.red
        btn.setTitle("button", for: .normal)
        btn.setTitleColor(UIColor.blue, for: .normal)
        btn.addTarget(self, action: #selector(ViewController.btnf(sender:)), for: .touchUpInside)
        self.view.addSubview(btn)
    }

    internal func btnf(sender: UIButton){

    }

    override func didReceiveMemoryWarning() {
        super.didReceiveMemoryWarning()
        // Dispose of any resources that can be recreated.
    }

}

|
```

図 4. 出力されたソースコード



図 5. 配置の比較画像

参考文献

[松村 09]松村慎,大久保洋介,武田智道,清水紘己,扇克至,里吉洋一,本末英樹: 絵で見てわかる Web アプリ開発の仕組み, 株式会社翔泳社,2015.
[安本 14]大津真, 池田尚志:Xcode ではじめる Swift プログラミング, 株式会社ラトルズ, 2014.