

1. 担当 PM

五十嵐 悠紀

(明治大学 総合数理学部 先端メディアサイエンス学科 専任准教授)

2. クリエータ氏名

大淵雄生 (筑波大学 情報学群 情報メディア創成学類 2 年)

3. 委託金支払額

2,736,000 円

4. テーマ名

機械学習を活用してデザインからモバイルアプリのコードを自動生成するソフトウェア

5. 関連 Web サイト

プレビュー版配布サイト : <https://axstudio.dev>

6. テーマ概要

本プロジェクトは、プログラムを書かずにアイデアをそのままデザインしていくことでモバイルアプリコードを自動生成するシステムを開発した。本プロジェクトの特徴は、1) 従来コードが書けないと作成できなかったモバイルアプリケーションを、ユーザがコードを書かずに作ることができること、2) 自動出力されたコードは人間が読みやすく、コードを書ける人であっても最初のプロトタイプを作成するシステムとして使いやすいこと、3) ロジックを組んでいくことができること、などが挙げられる。

7. 採択理由

デザインとコードの対応関係に注目して、デザインを元にアプリケーションのコードを自動生成するソフトウェアの提案プロジェクトである。デザインに含まれていないコンポーネントの意味やレイアウト情報などは機械学習にて補う手法を提案している。プロトタイプもすでに実装し、現時点で 7~8 割の精度で動作していることを確認しており、より一般ユーザが簡便に使うために精度

向上を目指して手法の改善を行っていく。二次審査でのプレゼンテーションは非常にわかりやすく、ユースケースや、プロダクトの広め方、そしてそれを実装するための技術力に至るまで、PM 一同を納得させるものであった。持前のデザインセンスの良さを発揮しながら、これまでコードが書けなかったユーザ層が手軽にアプリを作ることができるように頑張りたい。

8. 開発目標

開発目標としては、アイデアがあれば、デザインをしていくだけでモバイルアプリが作れるようにするために、

- 誰でも使える UI を提供すること
- プログラミングの知識がなくとも使いこなせること
- プログラムとして出力してモバイルアプリを実行できること

などが挙げられる。これにより、プログラミング能力のない人でも、アイデアの実現を可能にし、プログラミングができる人であっても、アイデアを元に高速な開発が可能になる。

9. 進捗概要

本プロジェクトでは、デザインとコードの対応関係に注目して、デザインを元にモバイルアプリケーションのコードを自動生成するシステム「AxStudio」を開発した（図 1）。

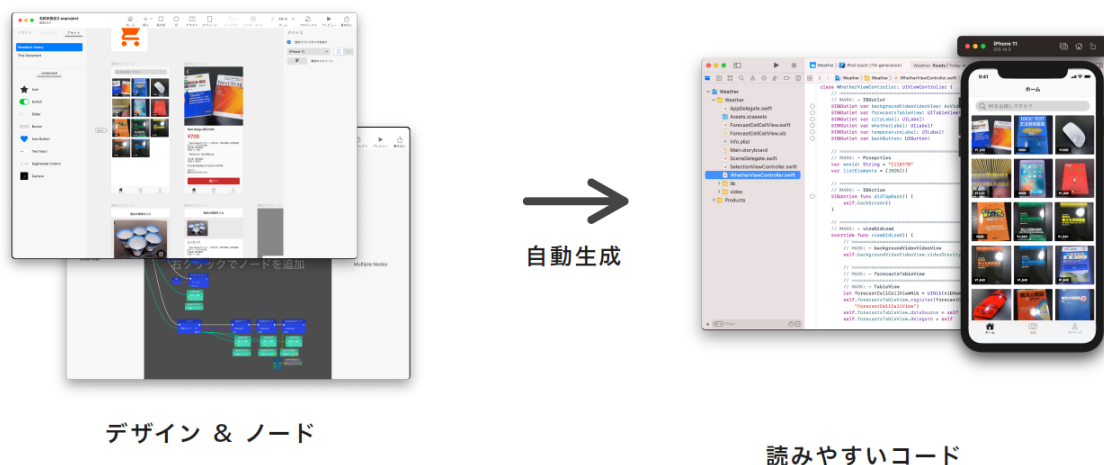


図 1：デザインとノードからプログラマに読みやすいコードを自動生成する「AxStudio」

AxStudio は既存のデザインソフトのようなユーザインタフェースになっており、ユーザはモバイルアプリに必要なデザインをすべて、AxStudio 内で行うことができるようになってきている。図 2 にデザインエディタの様子を示す。

AxStudio では、図 3 に示したように、ノードを用いてロジックを作成可能であり、Firebase をバックエンドに置いたサービスやローカルで動くサービスなど自由なアプリを開発可能となっている。ノードエディタは少ない工数でロジックを開発することができるため、プログラムを書くことができるエンジニアにとっても、プログラムを書く以上の速度でアプリを実装することができる。

通常、自動生成されるコードは人間が読むことを想定しておらず、その後に編集することはほぼ不可能である。AxStudio で生成されたコードは図 4 のように、人間が読みやすいコードとなっており、プログラマがあとから調整を行ったリコードを追加したりすることができるようになってきている。変数名などの識別子を周囲の状況から生成したり、不必要な構造を最大まで省いたりすることで美しいコードの自動出力を実現した。

本プロジェクトで開発した AxStudio を使って実際にいくつかのアプリを開発しているが、デザイン 10 分、ロジックに 10 分といった合計 20 分で、音声でチャットを行うモバイルアプリを開発することができた。以上のように、アイデアをそのまま誰でも簡単にモバイルアプリにできるシステムを開発できた。

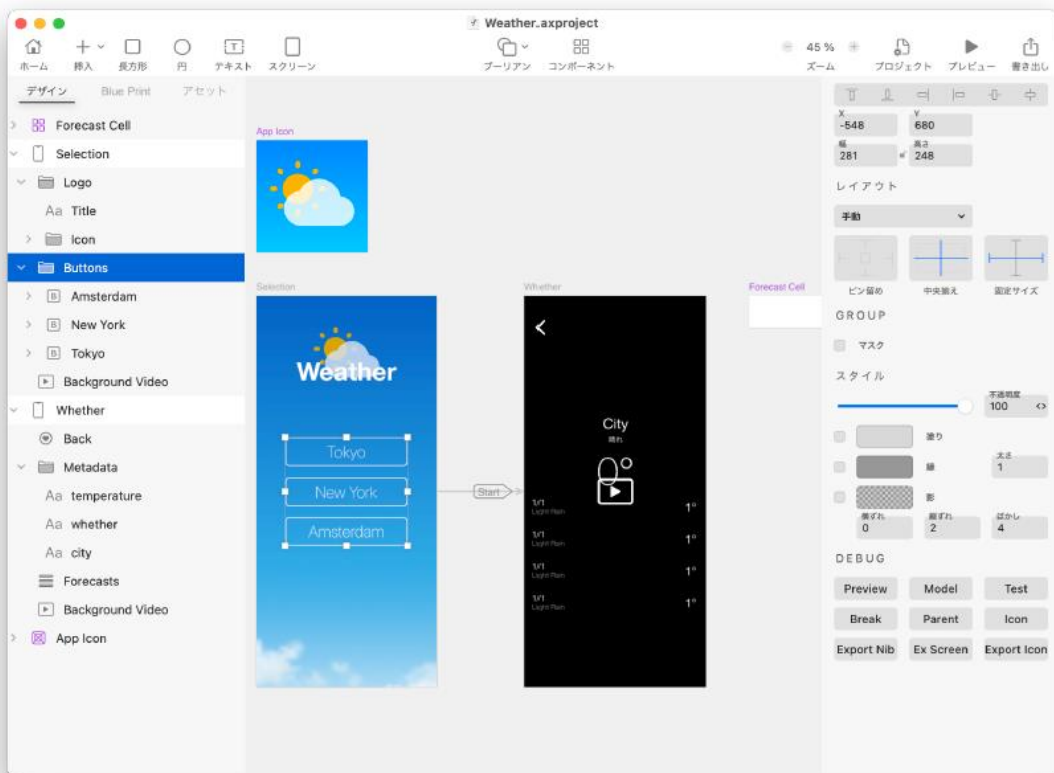


図 2 : AxStudio のデザインエディタ

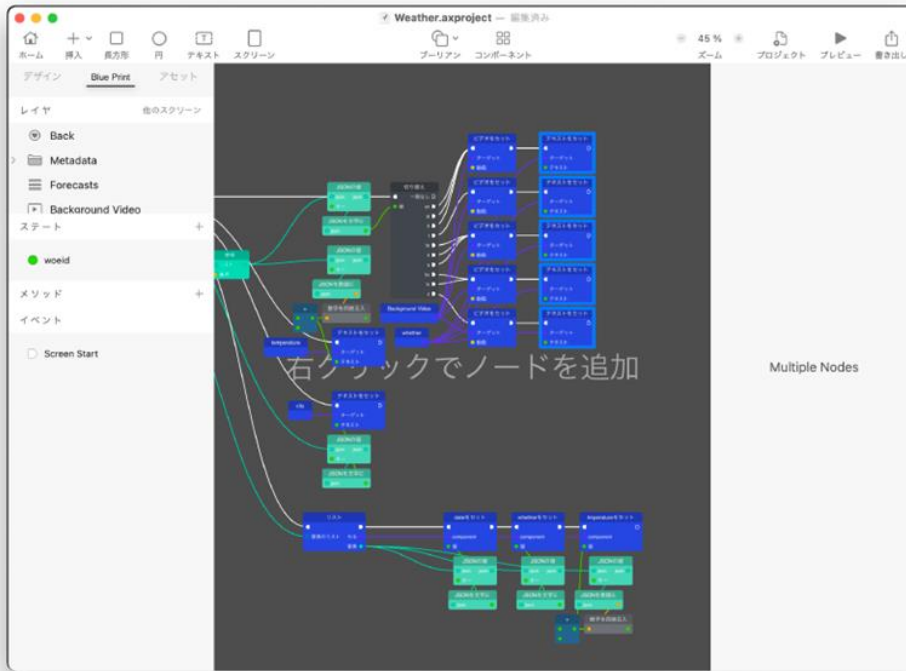


図 3 : AxStudio のノードエディタ

```

import UIKit

// MARK: - InitialScreenViewController
class InitialScreenViewController: UIViewController {
    // MARK: - IBOutlet
    @IBOutlet var textInputTextField: UITextField!
    @IBOutlet var buttonButton: UIButton!
    @IBOutlet var untitleTableView: UITableView!

    // MARK: - Properties
    // MARK: - Properties
    var todos: [String] = [] {
        didSet { self.todosDidSet(self.todos) }
    }

    var listElements = [String]()

    // MARK: - IBAction
    @IBAction func didTapButton() {
        self.todos.append(self.textInputTextField.text ?? "")
    }

    // MARK: - Methods
    // MARK: - Methods
    func todosDidSet(_ todos: [String]) {
        self.listElements = todos
        self.untitleTableView.reloadData()
    }

    // MARK: - viewDidLoad
    // MARK: - viewDidLoad
    override func viewDidLoad() {
        // MARK: - buttonButton
        // MARK: - buttonButton
        self.buttonButton.layer.cornerRadius = 3

        // MARK: - TableView
        // MARK: - TableView
        let listCellNib = UINib(nibName: "ListCellCell", bundle: .main)
        self.untitleTableView.register(listCellNib, forCellReuseIdentifier: "ListCellCell")
        self.untitleTableView.dataSource = self
        self.untitleTableView.delegate = self
    }
}

// MARK: UITableViewDataSource, UITableViewDelegate
// MARK: UITableViewDataSource, UITableViewDelegate
extension InitialScreenViewController: UITableViewDataSource, UITableViewDelegate {
    func tableView(_ tableView: UITableView, numberOfRowsInSection section: Int) -> Int {
        return self.listElements.count
    }

    func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) -> UITableViewCell {
        let cell = tableView.dequeueReusableCell(withIdentifier: "ListCellCell", for: indexPath) as! ListCellCell
        let element = self.listElements[indexPath.row]
        cell.テキストText = element
    }
}

```

図 4 : AxStudio によって自動出力されたコード

10. プロジェクト評価

本プロジェクトは、プログラミングができないユーザであっても、モバイルアプリをデザインして開発することができるソフトウェアの開発であった。当初は外部のデザインソフトウェアを使用してデザインを行い、それらによって生成されたファイルからコード生成をする機構を構築していた。その後、AxStudioにそれらデザインソフトウェアと同等の機能を持たせることで、AxStudio内でデザインからコード生成まで完結させることを可能にした。ここへさらに追加で、ノードベースでロジックを組むことができるようにすることで、プログラミングをすることができるエンジニアなどもユーザに取り込むことに成功した。AxStudioによって自動出力されるコードは可読性が高く、ユーザがアプリ作成の際に、ノードベースでロジックを組んだ上でコードへの加筆もできるシステムへと発展していった。これらの機能は、大淵氏が合宿やミーティングなどで多くのフィードバックを得ながら、どのような機能が必要かを見極め、その高い開発能力を発揮しながら開発を進めることで実現された。

AxStudioのプレビュー版配布後には、チュートリアルが欲しいとのユーザからの要望をうけてWebページへチュートリアルを整備するなど、一般ユーザが使い始めやすくする工夫をした。その後もユーザにどのように使われているかを調査し、AxStudio内でのプレビュー機能をさらに開発した。AxStudioのダウンロード数は2021年3月初旬の時点で1200を超えている。

このように、本プロジェクトの開発成果として、1つのソフトウェアの中で、デザイン、ロジック構築、ソースコードの自動生成、プレビュー画面での確認までを扱うことが可能となった点は評価に値する。

11. 今後の課題

本プロジェクトは、アイデアがある新規ビジネスを立ち上げたいユーザや、プロトタイプをさっと作りたいエンジニアといった想定してきたユーザ層以外にも多くの可能性を秘めている。一方、小学生にも試しに使ってもらったところ、アイデアをデザインとノードベースでアプリを作り、自動生成のコードを見て、プログラミングコードをどのように書くかを学んでいる様子が見て取れていた。このようなことを踏まえると、低年齢児のプログラミング教育にもつながるような展開も考えられそうである。このようなことから、事業化に向けての準備と社会への普及を期待している。