

## UML 要求仕様からの Android-UI プロトタイプ自動生成手法

松井 駿介<sup>†</sup> 小形 真平<sup>‡</sup> 松浦 佐江子<sup>‡</sup>芝浦工業大学 システム工学部 電子情報システム学科<sup>†</sup>芝浦工業大学 大学院工学研究科 機能制御システム専攻<sup>‡</sup>

## 1. はじめに

システム開発では、要求分析フェーズにおいて、開発者と顧客の両者が、開発するシステムの共通理解を正確に得る必要がある。それらの方法として主に用いられるのが、ソフトウェアの一部を作成する、プロトタイピング手法である。

我々はこれまで、UML(Unified Modeling Language)モデルで記述した要求仕様から、顧客も開発者も理解しやすい最終製品を模した HTML 形式の Web ユーザインターフェース(UI)プロトタイプを自動生成する手法[1]を提案してきた。これは、要求仕様とは独立したプロトタイプ開発に比べ、UML によって分析した要求内容をプロトタイプに正確に反映できるという利点がある。

しかし、当然のことながら、システム開発は Web-UI に限ったことだけではなく、それ以外のシステム開発の場合においても、プロトタイプを作成し、要求を明確化することは非常に重要なことである。

本研究では、現在急速な勢いで社会に浸透しつつあるスマートフォン(Android)のアプリケーション開発のプロトタイプ生成について考える。研究室内で行われた演習での Android 図書管理システムの開発事例に従来手法を適用し、「2. 問題定義」で述べる問題を解決する拡張手法を提案する。

## 2. 問題定義

従来手法では、プロトタイプで確認すべき内容として以下を挙げている。

- 業務手順
- 入出力項目

顧客は、開発されるシステムがどのような手順で業務を行うのか、そして、その手順を行う際の入力に対する出力結果、1 画面中の入出力項目に不足がないか、ということを確認できる。

これらの内容を実現させるために、UML モデルとしてアクティビティ図、クラス図、オブジェクト図を使用している。

アクティビティ図は、ユーザ、システム、(ユーザとシステム間の)インタラクションの 3 つのパーティションにおける処理の流れ、振る舞いを定義する。システム内のデータ構造はクラス図で作成し、1 画面中の入出力項目の確認という観点で、Input クラス(ユーザがシステムにどのような要素を入力するのか)、Output クラス(システムがユーザにどのような要素を出力するのか)を定義する。定義したデータ構造は、オブジェクト図に

Android-UI Automatic Prototyping From UML Requirement Specifications

<sup>†</sup>Shunsuke Matsui <sup>‡</sup>Shinpei Ogata <sup>‡</sup>Saeko Matsuura

<sup>†</sup>Department of Electronic Information System, Collage of System Engineering, Shibaura Institute of Technology

<sup>‡</sup>Division of Functional Control Systems, Graduate School of Engineering, Shibaura Institute of Technology

よって具体値を定義することができる。

Android 図書管理システムを扱う場合においても、前述の「業務手順」、「入出力項目」は考慮すべき確認内容であり、従来手法を扱うことで、要求の明確化が図れると考えた。

しかし、「画面(表示)の単位」という点において問題が起きた。スマートフォンが従来の PC に比べ狭小な画面空間であるため、画面の入出力項目が多く、操作がしづらい、見づらいという問題が起きた。

また Android では、表示させる内容に合わせて様々な UI 表示方法(「3.2. UI 表示方法」で説明)を扱うことができるが、従来手法は、1 つの表示形式に固定しており、表示される内容に対する重要度などの具体的な要件の意味づけは行われていない。

UI 表示方法について、Android 開発者向けサイト Android Developers の開発ガイド「User Interface」の項目内でも重要であることが記述されている[2]。

## 3. 提案手法

## 3.1. アクション

アクティビティ図内のアクションは、「目的語+を+動詞」の形に統一し、動詞は、UI として確認できるユーザとインタラクションのパーティションにおいて、あらかじめ定義されたものを扱うこととする。表 1 は、各パーティションにおいて、要求分析で記述されるべき機能要件の動詞とそれによって表現されるプロトタイプの UI 要素を定義したものになっている。

表 1 アクションの動詞と対応する UI 要素

パーティション	動詞	UI 要素
ユーザ	入力する	テキストボックス
	単数選択する	ラジオボタン
	複数選択する	チェックボックス
	選択する	リストビュー
	実行する	ボタン
インタラクション	表示する(要求する)	テキスト大表示

上記の表を用いて、アクティビティ図を記述することにより、アクティビティ図内の処理内容とプロトタイプが扱う処理を関連付けることが可能になる。また、アクションの動詞に対する処理の曖昧さをなくし、語句の統一を行なえるという利点もある。

## 3.2. UI 表示方法

Android では、Android-SDK によって提供されているアクティビティと呼ばれるクラスをベースとして、様々な UI 表示方法が用意されている。(表 2)

表示させる内容に合わせて扱うことにより、画面(表示)単位を細かく分割することや、表示される UI 要素に対して、各 UI 表示方法がもつ使用条件を意味づけることが可能になる。

本手法では、下記の表にある UI 表示方法をアクティビティ図にてステレオタイプで分類する。

表 2 指定できる UI 表示方法

名前	概要と使用条件	ステレオタイプの定義
アクティビティ	・Android の画面に相当する。UI として表示されるものは、必ずこのアクティビティが土台となる。 ・一般的な表示として使用する。	*ステレオタイプを設定しない場合に、UI 表示方法のデフォルトとして表示される
ダイアログ	・アクティビティの上に表示される小さなウィンドウ。 ・ユーザへの通知として使用する。	アクションに設定する <<dialog>>
トースト	・一時的に表示され、時間がたつと自動的に消える。 ・表示内容の重要性が低い、短いメッセージをユーザに通知するときに使用する。	アクションの表示する(要求する)に設定する <<toast>>
オプションメニュー	・Android のメニューボタンを押すと表示される。 ・補助的な機能を割り当てるときに使用する。	Input クラスに設定する <<OptionsMenu>>
コンテキストメニュー	・UI 要素を長押しすることで表示される。 ・UI 要素に対して、補助的な機能を割り当てるときに使用する。	Input クラスに設定する <<ContextMenu : UI 要素>>

### 3.3. 条件分岐

本手法のプロトタイプは UI に特化したプロトタイプであるため、システムパーティションの処理は表示されない。しかし、システムの処理の流れにおいて、条件分岐により、UI の出力要素が変更されることについては考慮する必要がある。

システムパーティション内に条件分岐があった場合、それらのガードを取得し、ダイアログとして条件分岐があったことを通知する。プロトタイプ利用者はダイアログで表示された条件分岐を選択し、その条件にあった出力結果を確認することができる。また、それが通常のダイアログと混同してしまわないように、条件分岐ダイアログとして、ダイアログの色を赤色に変更し表示している。

### 3.4. UML モデルからのソースコード生成

UML モデルで記述した内容から、前述の規則に従いソースコードを自動で生成する。生成するソースコードを以下に示す。

- アクティビティを記述した Java ファイル
  - 1つのアクティビティに対して、1つの Java ファイルを作成する。アクティビティは(インタクレーションを経由して)システムの出力に対するユーザの入力までを1つと考える。また、その他 UML モデルの記述内容に応じて以下を記述する。
    - ・アクティビティ以外の UI 表示方法
    - ・画面遷移
- 画面の要素、レイアウトを記述した Xml ファイル
  - Java ファイルで定義した1つの画面(表示)単位に対しての画面の要素、レイアウトを記述する。これには画面の入出力項目として定義した Input, Output クラスの情報を扱う。ただし、メニューで表示される要素は Xml ではなく Java のソースコード内で記述されるため、オプションメニューとコンテキストメニューで使用された Input クラスは記述しない。
- AndroidManifest.xml
  - アプリケーションの情報を記述するマニフェストファイルに、生成される Java のファイル名を記述する。
- styles.xml, colors.xml
  - 条件分岐ダイアログの色、スタイル情報を記述する。

前述以外のシステム起動に必要なファイル、フォルダ構成はすべて Android-SDK で提供されているコマンドを扱うことにより自動で生成することができる。

コンパイル、システム実行もコマンドから自動で行うことにより、UML モデルからのプロトタイプ自動生成を可能にする。

### 4. Android 図書管理システムへの本手法の適用

「3. 提案手法」で述べた本手法を Android 図書管理システムに適用する。

図 1 は、アクティビティ図「ISBN を直接入力する」の従来手法と本手法の違いを示している。

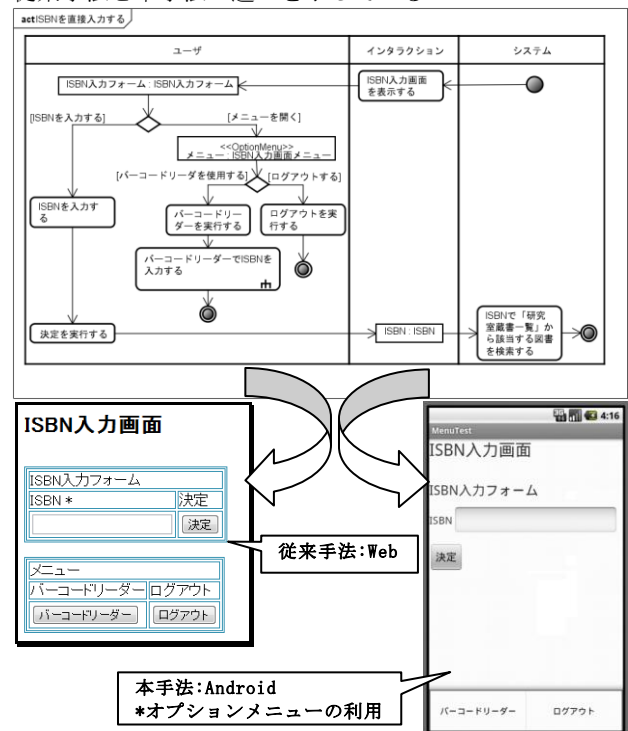


図 1 従来手法と本手法の違い

### 5. 考察とまとめ

本研究では、UML モデルから UI 表示方法を選択できる Android-UI のプロトタイプを自動生成することによって、Android の画面構成を考慮した要求分析を行うことができ、また UI 表示方法の指定により、表示させる内容に対して具体的な意味づけが可能になった。

しかし、Android には本研究で用いたもの以外にも多様な UI 表現方法があり、それらをプロトタイプで確認したいという要求も考えられる。

具体的な例として、アクションの「実行する」に対し、ボタンによる画面遷移もあれば、画面を指でスライドすることによって行うこともできる。

今後は、要求分析フェーズにおいて、多様な UI 表現方法を試すことができる Android プロトタイプ生成手法を提案していく。

### 6. 参考文献

- [1]小形, 松浦: UML 要求分析モデルからの段階的な Web UI プロトタイプ自動生成手法, 日本ソフトウェア科学会, コンピュータソフトウェア, Vol. 27, No. 2, pp. 14-32, 2010.
- [2]Android Developers(開発ガイド)  
<http://developer.android.com/intl/ja/index.html>