3L-1

スマートフォン向けアプリケーション設計による モデル駆動開発手法

松井 浩司† 松浦 佐江子‡

芝浦工業大学 システム理工学部 電子情報システム学科†‡

1. はじめに

近年，急速に普及してきたスマートフォンは携帯電話とも PC とも違う次の特長を持っている．

- センサ・GPS 等の多様な入力手段がある．
- タッチパネルによる高度な操作性を実現できる．
- 外部アプリケーションとの連携によりサービスの多様化，効率化が期待できる．

入力手段やアプリケーション連携はサービスの機能要件を決定する上で検討すべき項目であり，機能要件を実現する GUI(Graphical User Interface)の操作性もサービスの成否を決める重要な要素である．つまり，開発早期に入力手段および連携するアプリケーションとの関連性を明確にする，実装に近い水準で操作性を設計する必要がある．

MDD (Model Driven Development) は，OMG(Object Management Group)が提唱しているソフトウェア開発手法であり，UML (Unified Modeling Language)で記述した PIM(Platform Independent Model)を中心に開発を行い，PIM から PSM(Platform Specific Model)に変換し，PSM からソースコードを自動生成する開発手法である．システム再構築の容易化・ソースコード自動生成による効率化が特長であり，プラットフォームの進化が早いスマートフォンのアプリケーション開発に対して有効なソフトウェア開発手法である．しかし，GUI 開発はプラットフォームに依存しやすく，UML には抽象度の高い GUI のモデリング方法がない．そこで操作性に関する設計は，画面設計ツールである GUI ビルダを使用することが多いが，GUI ビルダは言語に依存したツールであり，抽象度の高い GUI 設計はできない．

本研究では，スマートフォンに依存したモデルと，スマートフォン OS に依存しない程度の抽象度で画面設計を行う GUI ビルダを用いて，言語依存の GUI を実現したスケルトンコードに変換し，実行することで，その操作性を早期に検証するモデル駆動開発手法を提案する．

2. 提案手法

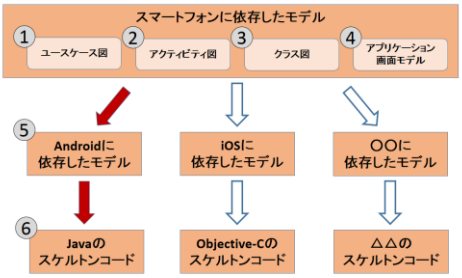

図 1 提案手法概要

本研究の提案手法は 6 つのステップで構成される．図 1 は，提案手法の概要および流れを示す．開発者が作成するのは，スマートフォンに依存したモデルである．①から③は UML，④は GUI ビルダを用いてモデルを作成する．⑤および⑥は，自動変換/生成による工程である．各ステップの説明で扱うモデル図は，本研究室で開発した「休講ナビ」のモデルである．休講ナビは，本学のホームページに掲載されている休講・補講情報を取得し，検索結果を表示する Android アプリケーションである．

2.1. スマートフォンに依存したモデル

スマートフォンに依存したモデルは，ユースケース図・アクティビティ図・クラス図・アプリケーション画面モデルから構成される．Android(4.0.3)，iOS(5.1.1)，Windows Phone(7.1.1) の SDK(Software Development Kit)からウィジェット・センサ等の共通機能 [1]を抽出し，それらをスマートフォンアプリケーションの特長を纏めた用語集とし，各モデルの要素を記述する．

2.2. ユースケース図の作成

ユースケースと連携する入力装置・アプリケーションがある場合，アクターとして記述する．表 1 にアクターの候補となる入力装置を示す．

表 1 アクターの候補一覧

アクター											
			センサ							カメラ	
ユーザ	外部アプリケーション	データベース	加速度センサ	方位センサ	明るさセンサ	近接センサ	ジャイロセンサ	GPS	Bluetooth	フロントカメラ	バックカメラ

図 2 は，休講ナビのユースケース図である．表 1 より，ユーザは学生に対応し，データベースは Web ページに対応する．

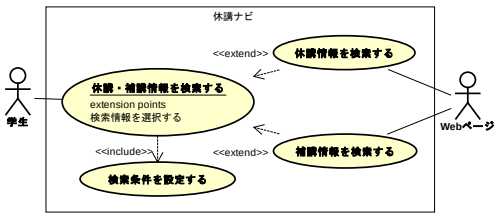

図 2 ユースケース図

2.3. アクティビティ図の作成

手続きを記述する際，アプリケーションに対するユーザの操作・ユーザに対するアプリケーションの表示・アプリケーション内部の処理に分割し，必ず手続きの対象となるオブジェクトを記述する．操作・表示の手続きの対象となるオブジェクトは，独自に定義した抽象ウィジェットを用いる．抽象ウィジェットは，入力要素・出力要素から構成され，ボタン・ラベル等のウィジェットを抽象的に表現したオブジェクトである．アプリケーション内部の処理に関する手続きを記述する際，パーティションを用いて，スマートフォンアプリケーションのライフサイクルからフォアグラウンド・バックグラウンドに分けて記述する．図 3 は，「休講・補講情報を検索する」のアクティビティ図である．学生には操作，インタ

Model Driven Development with Smartphone Application Design

†Koji Matsui ‡Saeko Matsuura

†‡Department of Electronic Information System, Collage of System Engineering, Shibaura Institute of Technology

ラクションには表示，フォアグラウンド・バックグラウンドには内部処理を記述する．ここで「検索条件を設定する」，「休講情報を検索する」，「補講情報を検索する」は，サブアクティビティであり，それぞれ別のアクティビティ図で手続きが記述される．これにより，すべての操作手順が定義される．

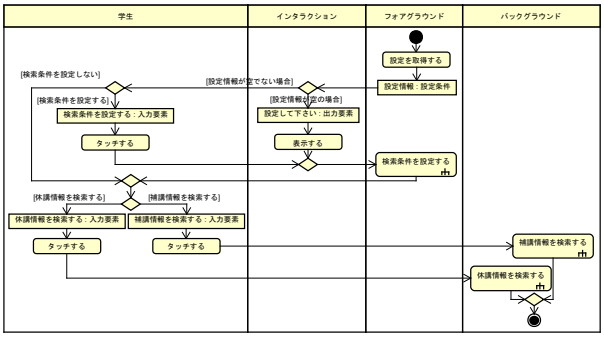

図 3 アクティビティ図

2.4. クラス図の作成

アクティビティ図に記述されたオブジェクトに対して，各ユースケースの入力要素・出力要素・エンティティデータをクラス図で定義する．ここで属性の型・ステレオタイプによって，入力および出力要素に対応する抽象ウィジェットに特長を付ける．ステレオタイプの種類は，表 2 に示す．図 4 は，「休講・補講情報を検索する」のクラス図である．抽象ウィジェットが保持する属性名は，ウィジェットの役割を表す．画面クラスは，アクティビティ図そのものに対応するクラスであり，入力要素・出力要素および全てのクラスを包括するクラスである．

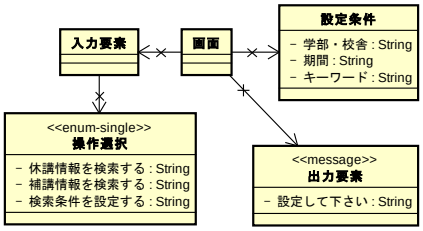

図 4 クラス図

2.5. アプリケーション画面モデルの作成

この工程は，すべてのアクティビティ図およびクラス図を入力とし，GUI ビルダを用いて画面レイアウト・画面遷移・画面内遷移を定義する．

表 2 抽象ウィジェットの変換ルール

抽象ウィジェット	入力要素				出力要素					
ステレオタイプ	<none>	<none>	<enum-single>	<enum-free>	<none>	<none>	<none>	<none>	<message>	<enum>
型	Stirng	int	String	String	String	JPG	MP3	Panel	String	String
ウィジェット	ボタン	スライダー	リストピッカー	リストピッカー	ラベル	イメージビュー	メディアプレイヤー	マップ	ダイアログ	リスト
	スイッチボタン	デートピッカー	ラジオボタン	チェックボックス				ブラウザ	トースト	
	テキストボックス	タイムピッカー	タブ							

GUI ビルダに入力された抽象ウィジェットは，表 2 に則して，複数のウィジェットに変換される．開発者は，変換された複数あるウィジェットから 1 つ選択し，各クラスの定義の基に画面設計を行う．図 4 の操作選択クラスが入力された場合，リストピッカー・ラジオボタン・タブに変換され，開発者は 1 つ選択し，GUI ビルダによって位置・大きさ等を定義する．図 4 の出力要素はダイアログとして設計する．更にアクティビティ図に現れるサブアクティビティに対応する画面の要素に対して，レイアウト・画面遷移・画面内遷移を決定する．つまり，アプリケーション画面モデルは，アクティビティ図を可視化したモデルともいえる．図 5 は，図 3 と図 4 および，サブアクティビティで表されたアクティビティ図と対応するクラス図を入力としたときに生成されたアプリケーション画面モデルである．アクティビティ図上に引かれた線の色と画面の枠線の色は対応しており，各アクション時の画面の状態を表している．枠線緑の画面は，サブアクティビティ内(検索条件を設定する)での画面を表しており，黄色矢印は画面遷移，水色矢印は画面内遷移を示す．

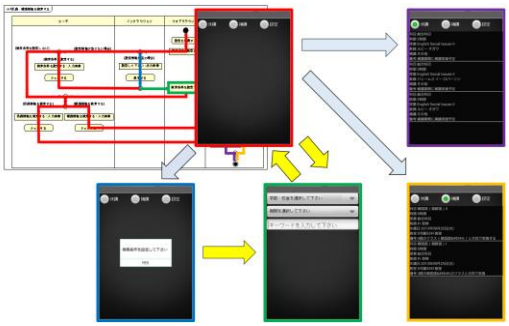

図 5 アプリケーション画面モデル

2.6. OS に依存したモデルへ自動変換

この工程は，スマートフォンに依存したモデルを，スマートフォン OS に依存したモデルに変換する．具体的には，抽象ウィジェットの名称をプログラミング言語に依存した名称に変換する．例えば，Android に依存したモデルに変換する場合，リストピッカーは Spinner，iOS では UIPickerView に変換される．

2.7. スケルトンコードの自動生成

この工程は，スマートフォン OS に依存したモデルからスケルトンコードを生成する．スケルトンコードの自動生成は，既存のツールを使用する．

3. まとめと今後の方針

本研究は，スマートフォンの特長を活かした MDD 手法を提案した．MDD において GUI 設計は重要であるが，井上尚紀ら [2]が提案する手法では，PIM に GUI 情報を組み込んではいるが，画面遷移や画面内遷移を考慮していない，といった問題があった．そこで本研究は，アクティビティ図を画面遷移および画面内遷移と対応させることで，画面制御と内部処理の整合性を保障しつつ，柔軟な画面設計を行うことができた．またモデリングの段階から各スマートフォン OS の SDK の共通機能を機能要件とすることで，動作環境となるスマートフォン端末を考慮したモデリングを行うことができた．

今後の方針としては，

- クラス図において，センサ等の入力装置の定義・変換ルールを検討する．
- iOS・Windows Phone でも同様の手法で，スケルトンコードが生成できるか検証する．

とする．

4. 参考文献

[1] 八木俊広，原昇平 ，かわかみひろき，実践 スマートフォンアプリケーション開発 iOS，Android，Windows Phone の比較，株式会社オライリー・ジャパン，2012.

[2] 井上尚紀 ，岸知二，“GUI を考慮した MDA 開発手法の提案,” 情報処理学会，2011