

スマートフォンアプリケーション設計に特化した UML および GUI ビルダによる相互的なモデリング手法

松井 浩司[†] 松浦 佐江子^{†‡}

芝浦工業大学 理工学研究科 電気電子情報工学専攻^{†‡}

1. はじめに

モデル駆動開発のベースモデルとして UML(Unified Modelling Language)モデルが代表的である。しかし、UML は内部設計には向いているが、画面構造・タッチパネル等による操作性等の外部設計をモデル化することは困難である。理由としては UML には外部設計に特化したモデルが存在しないからである。また GUI を考慮したモデル駆動開発手法にクラス図(UML モデル)を用いて外部設計を行う研究[1][2]があるが、以下の問題点がある。

1. Widget の座標・大きさ等をクラスの属性として定義するため、クラス図が肥大化し可視性が低下する。
2. 画面構成を直感的に設計・理解することができない。
3. スマートフォン特有の操作性の設計をしていない。

そこで我々は、各スマートフォン OS の共通機能を纏めた用語集に基づいた UML および GUI ビルダによって作成した「スマートフォン OS に依存しないモデル」をベースとしたモデル駆動開発[3]を提案してきた。我々が提案する手法では Widget の配置指定等の具体的な外部設計は UML では行わず、タブレット用アプリケーションとして開発する GUI ビルダを用いて直感的な画面設計およびスマートフォン特有の操作性設計を行う。また各ツールの長所を活かせるように各ツールで扱う設計情報を整合することでモデルの可視性の低下の防止を図った。しかし、異なるモデリングツールを用いて整合性のあるモデルを作成することは難しい。そこで本稿では異なる 2 つのツールで作成する「スマートフォン OS に依存しないモデル」における各ツール間で共通する設計情報と固有の設計情報について、GUI ビルダによるモデルの変更が与える UML モデルへの影響の観点から事例を用いて報告する。

2. 提案手法の概要

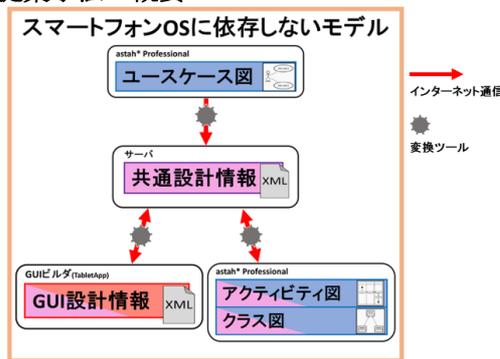


図 1 提案手法の概要

図 1 に提案手法の概要を示す。本稿で使用するモデリングツールは、UML モデリングツールである astah* Professional (以下、astah) とタブレット用アプリケーション

Mutual modelling method by smartphone application design specific UML and GUI builder

[†]Koji Matsui [‡]Saeko Matsuura

^{†‡}Division of Electrical Engineering and Computer Science, Graduate School of Engineering and Science, Shibaura Institute of Technology

ョンとして開発する GUI ビルダの 2 つである。そして各スマートフォン OS の共通機能を纏めた用語集と各ツールで作成するモデル要素を対応付けることで「スマートフォン OS に依存しないモデル」を作成する。また UML モデルと共通設計情報の変換を自動化するために astah* Plug-in を開発する。「スマートフォン OS に依存しないモデル」の作成ステップを以下に示す。

1. astah を用いてユースケース図を作成する。
2. GUI ビルダによる外部設計または astah による内部設計(アクティビティ図及びクラス図の作成)を選択する。
3. ステップ 2 で選択したツールでモデリングを行う。
4. ステップ 2, 3 を繰り返す。

上記からユースケース図の作成後はツールの使用順序に制約はないことが分かるが、前述で述べた目的から、後述する事例では GUI ビルダ・astah の順でモデリングを行うとする。

2.1. 共通設計情報とは

内部/外部設計で扱う設計情報は完全に独立していない。例えば UML を用いて内部処理を記述する場合、処理のきっかけとなるトリガーの多くは画面を通したユーザの操作であるため、トリガーとなる Widget およびタップ等の操作方法は明らかにしなければならない。また GUI ビルダによって画面構成を設計する場合、画面を通してユーザに表示する結果などのエンティティデータは何らかの内部処理から得られるものであるため、本当にそのタイミングで期待するデータが生成できるか明らかになっていなければならない。そこで異なるツール間で共通する設計情報を図 2 に示す。

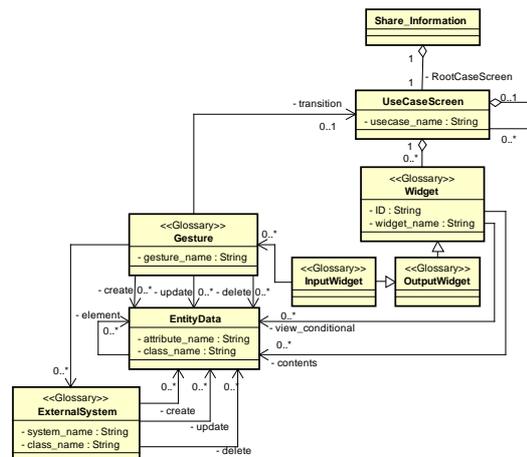


図 2 共通設計情報の定義

一部のクラスに付加されているステレオタイプ(Glossary)は用語集と対応している。内部設計の観点から図 2 を見ると、内部設計で扱う外部設計に関する設計情報は、ID(Widget を識別する文字列)と

widget_name(Button など)と Gesture(Tap など)であり、位置・大きさといった視覚的な設計情報は内部設計では扱わない。逆に外部設計の観点から図 2 を見ると、EntityData の CRUD に関する設計情報は扱うが、ある EntityData に対してどのような処理をするかといった実現可能性の検討に大きく関わる内部ロジックは扱わない。

以上のように各ツールの長所を活かせるように共通する設計情報を定義し、相互的なモデリング手法を目指す。

2.2. 事例

本稿で扱う事例は、本研究室で開発した本学の HP から休講/補講情報を取得し表示する「休講ナビ」を用いる。休講ナビのユースケース図とユースケース図から生成された共通設計情報を図 3 に示す。

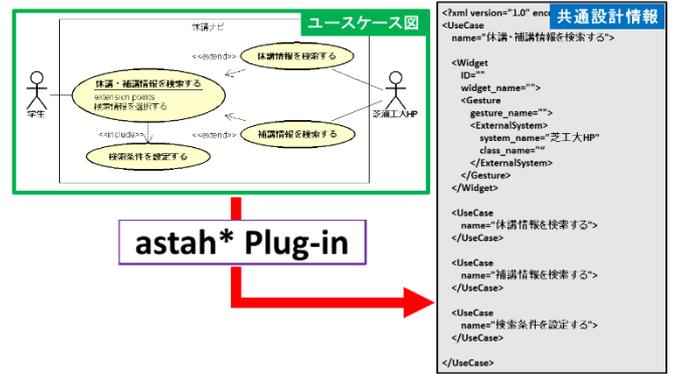


図 3 休講ナビのユースケース図と共通設計情報

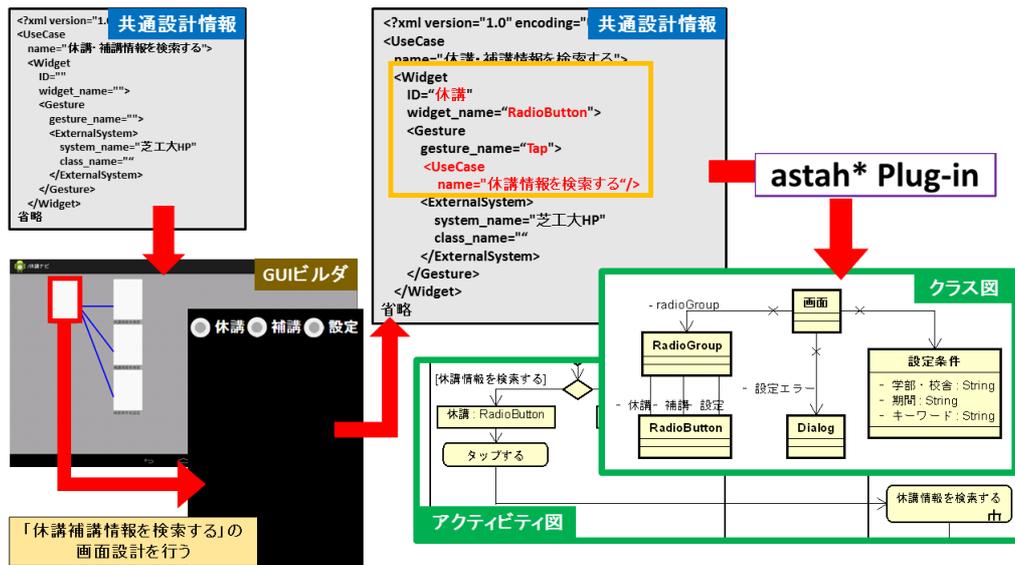


図 4 モデリング手順

図 4 は図 3 のユースケース図から生成した共通設計情報(図 4 左上の共通設計情報と同等)を GUI ビルダで読み込み、「休講/補講情報を検索する」についての外部設計を行い、外部設計によって更新された共通設計情報からアクティビティ図・クラス図を生成した流れを示している。初期状態の共通設計情報を GUI ビルダで読み込むとユースケースの数だけ空の画面が表示される。次に編集したい画面を選択し、図 2 で指定した誘導可能性の方向に沿って定義を行う。まず Widget を選択し、Widget の位置等を指定する。選択した Widget が InputWidget であった場合、Gesture を決定し、その操作によって CRUD の対象となる EntityData の定義、連携する ExternalSystem の定義、遷移する画面の決定を行う。また Widget の表示 / 非表示は指定した EntityData(view_conditional)の何らかの条件(有無など)によって決定する。これにより例外の挙動を実現する。

GUI 設計情報から再び共通設計情報に変換すると図 4 の四角で囲まれた部分が更新され、この共通設計情報から内部ロジックなどが抜けた UML モデルが生成される。外部設計にて RadioButton に対して位置・大きさ等を定義しているが、クラス図にはそれら視覚的な設計情報は反映されず、GUI 設計情報が保持したままである。

3. まとめと今後の課題

GUI ビルダと UML で作成する「スマートフォン OS に依存しないモデル」における各ツール間で共通する設計情

報と固有の設計情報について、GUI ビルダによるモデルの変更が与える UML モデルへの影響の観点から事例を用いて報告した。外部設計を UML で行った従来研究[1][2]に対し、本研究ではタブレット用アプリケーションである GUI ビルダを用いることで直感的な画面設計およびスマートフォン特有の操作性設計を実現し、また各ツールで扱う設計情報を整合することで、モデルの肥大化を防ぎ可視性を向上させた。

しかし、GUI ビルダおよび自動化のためのツールは実装していないため、本稿が提案した定義に誤りがある可能性を認めない。そこで今後の課題として、GUI ビルダおよび自動化のための変換ツールを実装し、実際に「スマートフォン OS に依存しないモデル」を作成することで、定義の見直し改善を行う。

参考文献

[1] Ayoub SABRAOUI, Mohammed EL KOUTBI:GUI Code Generation for Android Applications Using a MDA Approach, ICCS, pp.1-6, 2012
 [2] G. Botturi, E. Ebeid, F. Fummi, D. Quaglia:Model-driven design for the development of multi-platform smartphone applications, FDL, pp.1-8, 2013
 [3] 松井浩司, 松浦佐江子: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, ソフトウェアエンジニアリングシンポジウム 2014, pp.208-209, 2014