

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^{†1} 松浦佐江子^{†2}

スマートフォンは、フィーチャーフォンや PC とは異なる 3 つの新しい特長を持っている。1 つ目はセンサ・GPS 等による多様な入力、2 つ目はタッチパネルによる高度な操作性、3 つ目は外部アプリケーションと容易に連携できる仕組みを持っている事である。それらの特長を活用する事で、革新的かつ便利なサービスを提供する事ができる。そこで本研究では、スマートフォンの高度な機能を活用したアプリケーションを開発するために、UML と GUI ビルダで作成するスマートフォンに依存したモデルを用いたモデル駆動開発手法を提案する。

MDD for Smartphone Application with Smartphone Feature Specific Model and GUI Builder^a

KOJI MATSUI^{†1} SAEKO MATSUURA^{†2}

Smartphone has a new three features that are different from PC and feature phones. First, there are a variety of input devices such as GPS and sensors. Secondly, advanced operability may be achieved by using the touch panel. Thirdly, the service can be enriched by easily connecting with external applications. By taking advantage of their characteristics, it is possible to provide convenient and innovative services. In this paper, in order to develop applications that take advantage of the advanced features of smartphone, we propose a model-driven development method using a Smartphone Feature Specific Model that a developer can design by using the GUI builder and UML.

1. はじめに

近年、急速に普及しているスマートフォンは PC ともフィーチャーフォンとも異なる以下の特長を持っている。

- センサ・GPS 等の多様な入力装置がある。
- タッチパネルによる高度な操作性が実現できる。
- 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。

入力装置やアプリケーションとの連携はサービスの機能要件を決定する上で、開発早期に検討すべき項目であり、機能要件を実現する GUI (Graphical User Interface) および操作性もサービスの成否を決める重要な要素である。特にタッチパネルによる操作性は実際に操作をしないと使い勝手が分からないため、実装に近い水準で操作性を検討する必要がある。

スマートフォンの高性能化に伴い、スマートフォンの活用範囲が広がり、ユーザからの要求は刻々と変化する。そのため、スマートフォン市場の流れは早く、それに対応するために、スマートフォン開発ではスピーディーな開発体系が求められている。しかし、スマートフォンの端末数が多いことから、複数のプラットフォームを考慮して開発する必要があり、開発者側の対応は難しくなっている。そこで注目されているのが MDD (Model Driven Development) である。MDD は、UML (Unified Modeling Language) などのモデルでソフトウェアの要求分析・設計を行い、モデルコンパイラによってモデルからソースコードを自動生成するソフトウェア開発手法である。以下のような、大きな利点があり、スマートフォンアプリケーション開発の効率化に有効である。

- ソースコードの自動生成によって時間的コストを削減できる。
- 他のプラットフォームでのサービス再構築が容易である。

2. 既存の開発方法の問題点

UML にはシステムの静的な構造や動的な振る舞いに関する内部設計を定義するモデルは用意されているが、GUI を定義することに特化したモデルは存在しない。そのため GUI 設計においては GUI ビルダ・モックアップなどのツールを用いて行うことが一般的である。しかし、それらのツールはプログラミング言語に依存するため、他のプラットフォームで流用することが難しく、さらに PC 用のツールであるために、スマートフォン特有の操作性まで検討することが困難であるという問題がある。

また GUI ビルダは利用イメージから開発するため、サービスの全体像を把握し、その実現可能性を検討するという観点からの内部設計を行うことに向いていないため、内部設計は UML、外部設計は GUI ビルダのように問題領域によってツールを使い分けた方が効率的である。また開発するアプリケーションによっては内部設計から、または外部設計から開発を行った方が効率的というケースも考えられるのでツールの使い分けが流動的に行えたら、さらに効率的な開発体系になる。このような開発体系を実現するためには、各ツールが扱う問題領域の切り分けを明確に定義する必要がある。例えば内部設計に向いていない GUI ビルダだからといって、内部設計と 100%切り離して外部設計を行うことはありえない。いくらリッチで使い易い画面を設計しても、そこで表示するデータの存在が不確実なもので

^{†1†2} 芝浦工業大学
Shibaura Institute of Technology

あれば破綻する可能性がある。つまり各ツールの長所を損なわない程度にツール間で整合性が保証される仕組みを定義することが解決すべき問題である。

3. 問題解決へのアプローチ

本稿では、2章で挙げた問題解決を目的とした GUI ビルダを用いたスマートフォンアプリケーション向けモデル駆動開発手法を提案する。GUI 設計および操作性を開発早期に議論するためにタブレット上で実行する GUI ビルダを開発し、要求分析・アプリケーションの静的な構造・動的な振る舞いを定義するためにユースケース図・アクティビティ図・クラス図を用いる。各 UML モデルに必要最低限の GUI 情報を持たせることで、GUI ビルダと UML の短所をそれぞれのツールの長所で補う。また特定のプラットフォームに非依存なモデルを作成するために、Android・iOS・Windows Phone の SDK(Software Development kit)[1] から前述したスマートフォンの特長に関連し、かつ各スマートフォン OS に共通する機能のみを纏めた用語集を用意し、テンプレートとしてモデル要素を記述する。

4. 提案手法

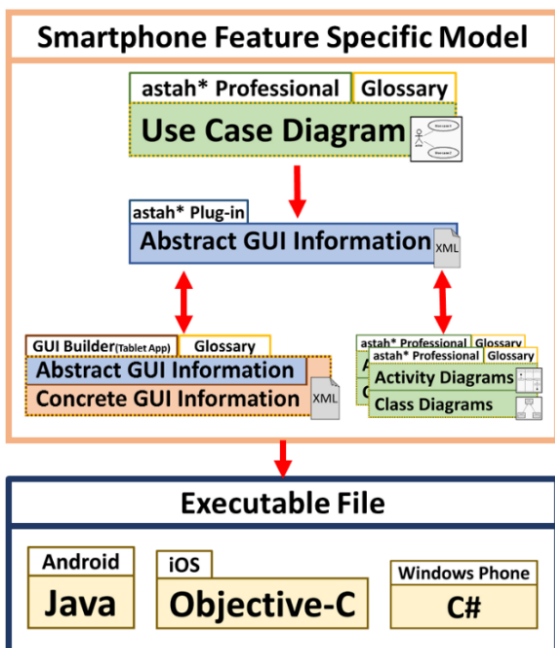


図 1 提案手法の概要

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) [2] と Android タブレット用アプリケーションとして独自に開発した GUI ビルダである。それらモデリングツールを用いて作成したモデルを総称してスマートフォンに依存したモデル (Smartphone Feature Specific Model) とする。スマートフォンに依存したモデルは、特定のスマートフォン OS に依存してはならない。そこで、各モデリングツールを使用する際には、特定のプラットフォームに依存することを防ぐため、前述したスマートフォン OS の共通機能を纏めた用語集 (Glossary) を用いてモデリングを行う。また GUI ビルダと astah 間の情報交換を自動化するために astah* Plug-in を開発する。

開発者は、astah でユースケース図を作成し、作成したユースケース図から astah* Plug-in を用いて抽象 GUI 情報 (Abstract GUI Information) を取得する。抽象 GUI 情報とは、UML モデル上での議論に必要な GUI 情報を削って得られる情報である。削る情報とは、ウィジェットの位置・大きさなどが該当し、これを具体 GUI 情報 (Concrete GUI Information) といふ GUI ビルダによって定義する。次に開発者は、外部設計から行うのか、内部設計から行うのかを自由に選択する。

外部設計から行う場合、抽象 GUI 情報を GUI ビルダに入力し、抽象 GUI 情報に対して画面構成・画面遷移などを決定する。外部設計から内部設計に移る場合には、抽象 GUI 情報のみから astah* Plug-in に通して、1つのユースケースに対応するアクティビティ図とクラス図のセットを作成する。つまり、アクティビティ図・クラス図はユースケース数だけ存在する。既に UML モデルが存在する場合は差分のみを更新する。アクティビティ図では処理の手続きと処理の対象となるデータを定義し、クラス図ではアクティビティ図で定義したデータに対して構造化を行う。内部設計から外部設計へ移る場合は、アクティビティ図・クラス図から抽象 GUI 情報を取得し、GUI ビルダへの入力とする。既に具体 GUI 情報が存在する場合は、抽象 GUI 情報と具体 GUI 情報の整合性を取り、外部設計を行う。また外部設計から内部設計へ、内部設計から外部設計へ移るタイミングに制約は設けない。最終的には、作成した抽象 GUI 情報・具体 GUI 情報・クラス図を基に各 OS で実行可能なスケルトンコードを生成する。

5. おわりに

本稿では UML と GUI ビルダを用いたスマートフォンアプリケーション向けモデル駆動開発を提案した。スマートフォンの特長を纏めた用語集を用いた特定のプラットフォームに依存しないモデルと抽象 GUI 情報という中間モデルを定義したことで GUI ビルダと astah 間で整合性を保ちつつ設計が進められ、UML による MDD の利点と、直感的な GUI ビルダの利点を活かした開発を行えるようになる。と考える。

今後の課題としては以下とする。

- GUI ビルダや astah* Plug-in などのツールを完成する。
- スマートフォンに依存したモデルからスケルトンコードを生成する際のマッピングルールを検討する。

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) astah* Professional, <http://astah.change-vision.com/ja/product/astah-professional.html>
- 3) スティーヴ J.メラー, ケンドール・スコット, アクセル・ウール, デイルク・ヴァイセ: MDA のエッセンス, 株式会社 翔泳社(2004)