3J-04

UMLモデリングツールとGUIビルダによる相互的なモデリング手法の実装と適応事例による評価

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

スマートフォンアプリケーションに対する要求の高度化に伴い，アプリケーション開発の大規模化，複雑化が顕著になっている．モデル駆動開発はその解決策として有望であり，ベースモデルとしてはUML(Unified Modelling Language)モデルが代表的である．しかしUMLは内部設計のモデル化には適しているが，詳細な画面を設計する外部設計のモデル化には詳細なレイアウト等を記述する必要があり，直観的な理解が難しいという問題[1][2]がある．そこで内部設計をUMLモデリングツール，外部設計には画面を可視化するGUIビルダで行うモデル駆動開発を提案してきた[3]．本稿では各モデリングツール間で共通する設計情報の定義に基づき双方のモデル間で矛盾なく共有できるGUIビルダを実装した．スマートフォンアプリケーションの事例開発を通じて相互的なモデリング手法の有効性を議論する．

2. 提案手法

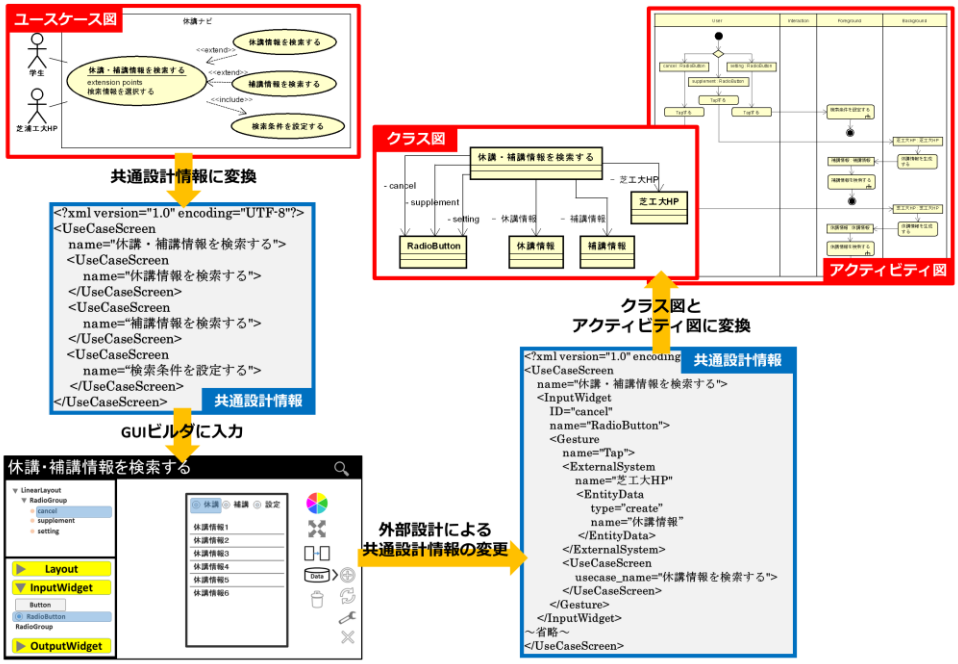

図 1 提案手法の適応事例（外部設計から内部設計へ）

図 1は「休講ナビ」を本手法に適応した事例である．休講ナビとは本学の HP から取得した休講・補講情報を，ユーザが設定した検索条件によりパーソナライズし，閲覧できるAndroidアプリケーションである．

本手法で使用するモデリングツールは astah* professional とタブレット用 Android アプリケーションとして開発したGUIビルダである．最初に開発者はユースケース図を用いてアプリケーションに要求する機能を定義する．各ユースケースは画面に対応し，関連は画面遷移順序に対応する．また最終的に生成するクラス図とアクティビティ図は各ユースケースのユースケース記述を表すアクティビティ図，ユースケースをコントロールするクラス図として同名で１つずつ生成する．アクティビティ図上のサブアクティビティは遷移先の画面に対応するユースケースである．本手法では，スマートフォンの特徴である外部システムとの連携やセンサーの使用について特化したUMLモデルの拡張メタモデル[3]に基づき内部設計を行うので，UML で定義した内部設計モデルから外部設計及び内部設計の両者の接点として必要な共通設計情報を生成し，GUI ビルダに入力し外部設計を行うことができる．逆にGUIビルダで設計した外部設計から共通設計情報を生成し，クラス図及びアクティビティ図に変換して内部設計を行うこともできる．図 1のユースケース図から生成された共通設計情報をGUIビルダに入力すると，各ユースケースに対応する画面とその遷移が初期状態となる．各画面に対し次に示すGUIビルダの主な機能を用いて画面の構成を設計する．

Implementation and evaluation of modeling method by mutual use of UML modeling tool and GUI builder
†Koji Matsui
‡Division of Electrical Engineering and Computer Science, Graduate School of Engineering and Science, Shibaura Institute of Technology

Copyright ©2016 Information Processing Society of Japan.
All Rights Reserved.

● Widget の配置
● 配置した Widget に対して位置，大きさ等の指定
● 画面遷移のトリガーとなる Widget の指定
● CRUD のトリガーとなる Widget の指定
● 画面の統合
● 上記 5 つで指定した項目の動作確認

GUI ビルダで配置できる Widget 等のスマートフォン特有のコンポーネントは Android，iOS，Windows Phone の共通機能を纏めた用語集に基づいている．画面遷移のトリガーは内部設計のアクションと，CRUD のトリガーとなる Widget の定義は内部設計のエンティティデータの CRUD アクションと結びつくため，共通設計情報として残し，内部設計の UML モデルに反映する必要がある．ここでは休講情報は本学の HP よりデータを取得するので外部システムからのデータの取得が必要であり，そのデータをエンティティデータ「休講情報」として内部設計する必要がある．一方，図 1 の RadioButton のレイアウトに関する設計情報や「休講情報を検索する」の画面の統合は共通設計情報には残さず，GUI ビルダ固有のデータとして保持する．最後に共通設計情報から，内部設計のメタモデルに従ってクラス図とアクティビティ図を生成し，それを基に内部設計を続行する．

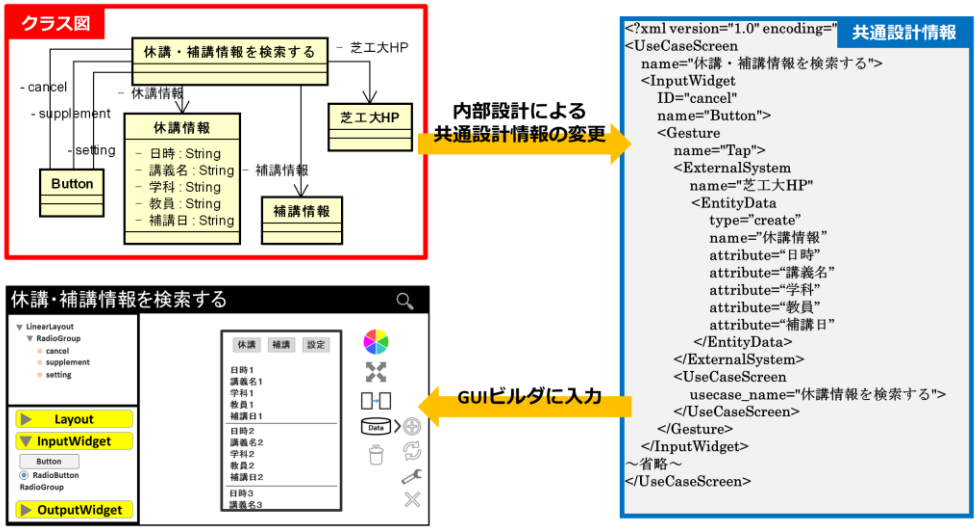

図 2 内部設計による外部設計の変更

図 2では，図 1で生成したクラス図内のエンティティデータである「休講情報」の属性を，HP データを基に定義し，再度 GUI ビルダで休講情報の画面出力を定義する過程を表している．

● RadioButton から Button へ変更
● 休講情報に「日時」「講義名」の属性を追加

図 1と図 2によりリスト内に表示される項目の変化は明らかである．また RadioButton から Button の変化も明らかであるが注目すべき点はレイアウトに一切の変更がないことである．上記で GUI ビルダによるレイアウトの変更は共通設計情報に影響を与えないと述べた通り，共通設計情報は Widget に関するレイアウトの設計情報は一切持たない．そのため，RadioButton が Button に差し替わるだけでレイアウトに変更は起こらない．

3. 考察

本稿では各モデリングツール間で共通する設計情報の定義に基づき双方のモデル間で矛盾なく共有できる GUI ビルダを実装し，スマートフォンアプリケーションの事例開発を通じて相互的なモデリング手法の説明をした．休講ナビのように表示項目が多いアプリケーションにおいてはクラス図で議論するより，想定している画面構成を可視化した状態で議論した方がユーザビリティの向上に繋がり，また既存のデータを取得し表示するといった単純なロジックのアプリケーション開発という観点においても開発早期から完成形のイメージを固め易いため円滑に開発を進めることができた．逆にデータ構造，ロジックについて深く議論すべきアプリケーション開発においても内部設計を行った後，動かない UML モデルも GUI ビルダを通してプロトタイプのように動作確認が可能なので内部設計を別の観点から議論することができた．

4. 今後の課題

本手法の GUI ビルダはプロトタイプのように動作確認が可能だが，本番環境でのデータ及びデータ量を想定し，再現することができない．例えばデータ量が多ければ，それに対応した見せ方も様々であるので実際のデータを考慮した画面設計を行えるようにしなければならない．

参照文献

[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

Copyright ©2016 Information Processing Society of Japan.
All Rights Reserved.