6L-01

UML 要求仕様における関心事の分離によるモデル駆動開発手法

川合 怜† 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

モデル駆動開発は多様な実装技術のソフトウェアを自動変換により効率よく開発できる有用な手段である．しかし，要求を試行錯誤して整理していくソフトウェア開発初期の段階では，データ構造や振る舞いは曖昧なものになることから，自動変換を適用できる程にモデルを厳密に定義することは難しい．

そこで，ユーザとシステムのインタラクションに顕在する機能要求の分析により，UI(User Interface)とシステムの内部ロジックにおける振る舞いとデータの関係を UML(Unified Modeling Language)モデルを用いて明確にする．そして，機能を正しく実現するために必要な制約を UML 標準の制約記述言語 OCL(Object Constraint Language)により定義する．このような機能要求の関心事の分離によるモデルの曖昧性を削減した UML 要求仕様を用いたモデル駆動開発手法を提案する．

2. 要求分析モデル

ソフトウェア開発初期に行われる要求分析で検討すべき機能要求の観点は，(1)ユーザの操作手順とその入力からなるシステムの応答を示す UI，(2)入力に対する応答を実現するためのシステムの内部ロジック，(3)機能を正しく実現するための振る舞いとデータの制約である．

(1)，(2)は要求分析モデル[1]により定義することができる．要求分析モデルでは，インタラクションに顕在する機能要求を明らかにするため，UML を用いたユースケース分析により，振る舞いの流れとデータ構造を定義する．具体的にはユースケースにおける振る舞いの流れ，機能の実行手順を表すアクション系列の基本フロー，パーティションによる主体者と振る舞いの関係，オブジェクトノードによるデータと振る舞いの関係，ガードの明記による例外フローをアクティビティ図で定義する．また，インタラクションにおける入出力とシステム内部のデータ構造をクラス図で定義する．

要求分析モデルでは，ユースケース毎に基本フローにおける振る舞いやデータにより発生し得る例外フローの条件を定義していくが，複数のデータ間の関連も加味して定義するため，フローが複雑になる恐れがある．また，ユースケース間で条件の不整合が起こる可能性がある．機能を実現するためには，様々なユースケースから不整合なくクラスの操作を切り分ける必要があるため，要求分析モデルでは(3)を適切に定義することが困難であり，誤りが入りやすい．

そこで本研究では，クラスへの不変条件の定義とそれによる例外フローの条件の定義により，ユースケース間の整合性を保持できるように関心事の分離を行う．

Model Driven Development by Separating Concerns in UML Requirement Specification

†Satoshi Kawai ‡Saeko Matsuura

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

3. 提案手法

3.1. 概要

図 1 に提案手法の概要を示す．本研究では要求分析モデルの定義を出発点とし，そのモデルに対して機能を正しく実現するために必要な制約を OCL により定義する．そして，設計モデルへの変換，設計情報の追加をしていくことで最終的にソースコードを生成する．以下，各手順について事例を交えながら説明する．

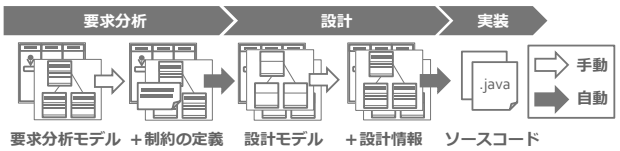

図 1 提案手法の概要

3.2. 制約の定義

データの不変条件は OCL を用いてクラス図に定義する．そして，アクティビティ図における例外フローでは，不変条件名をガードに明記することで分岐条件を明確にし，各ユースケースの条件の整合性を保つことができる．例外フローでは，その遷移先を定義することが重要な点の一つであることから，遷移先が同一な分岐条件を一つにまとめることで例外フローによる複雑化を回避する．

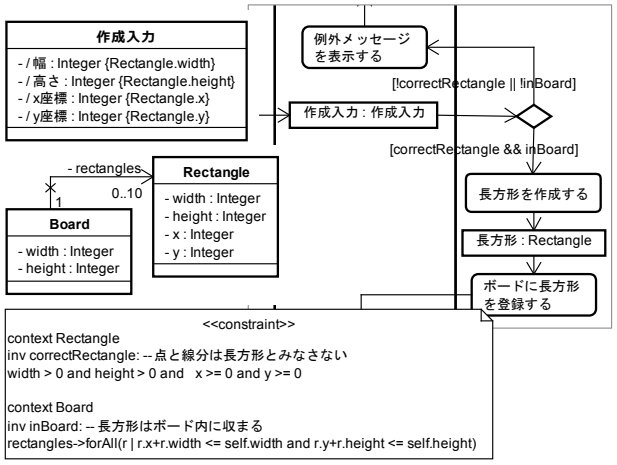

図 2 要求分析モデルへの制約の定義

図 2 は事例である長方形エディタにおける Rectangle クラスの不変条件「点および線分は長方形とみなさい」とボードの不変条件「長方形はボードに収まる」を定義し，アクション「長方形を作成する」で発生する例外フローの分岐条件を不変条件名により定義したものである．このように制約定義の関心事を分離して考えることで，各ユースケースにおける例外条件の不整合を起きないように管理することができる．

また，インタラクションで流れる入出力データは，システムの内部ロジックで扱う Entity データの派生データであることから，その関係を明確にするために OCL で定義を行う．

Copyright ©2015 Information Processing Society of Japan.
All Rights Reserved.

3.3. UML 要求仕様から設計モデルへの変換

設計では，機能を実現するためのアーキテクチャやフレームワーク，API の決定が行われる．そのため，設計モデルは様々な実装技術へ特化できるような構成が望ましい．本研究では，要求分析モデルに対して制約を定義したモデル(以降，UML 要求仕様）における関心事の分離と，インタラクティブな UI アプリケーションの実現を目的とした BCE(Boundary Control Entity)アーキテクチャパターンへのマッピングにより，フレームワークや API の導入を容易にする．図 3 に UML 要求仕様対するプロセスの分割と設計モデルとの対応関係を示す．

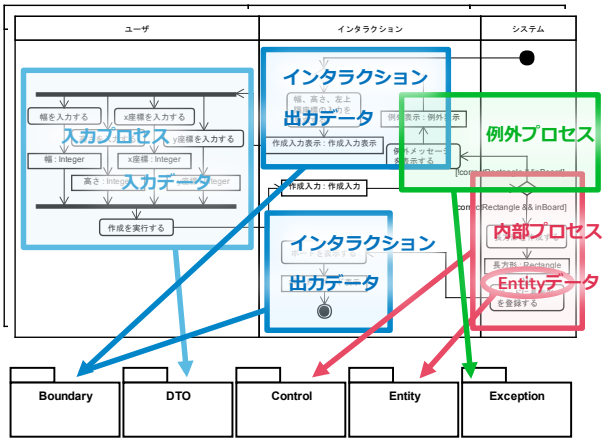

図 3　UML 要求仕様における関心事の分離と設計モデル

3.4. 設計モデルの詳細化とソースコード生成

フレームワークを決定した際，ユーザの入力操作とシステムの応答をどのような形式にするかを決める必要がある．例として，長方形エディタを Java Applet で設計を行う場合，図 3 の入力プロセスでは，ユーザが幅等を入力し，トリガを起点に内部プロセスを呼ぶことから，入力を TextField，トリガを Button のように入力形式を決定し，内部プロセスの呼び出しは，Button による定型的なイベント処理に基づいて実装される．また，TextField による入力値を整数型に変換する必要があることから，入力データを DTO(Data Transfer Object)クラスとして設計する．

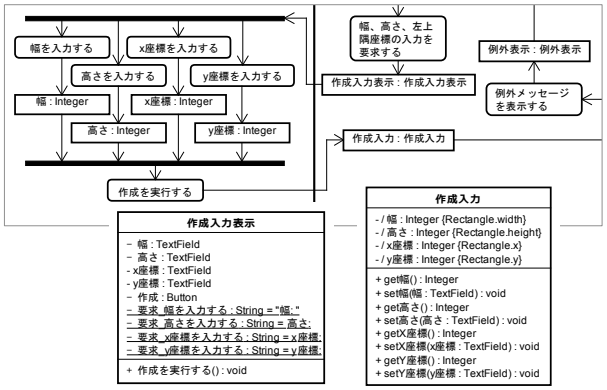

図 3　設計情報の追加によるクラスの詳細化

このように，関心事毎にプロセスやデータ構造を踏襲することで設計およびソースコードの生成ができる．

4. 考察

事例では，CUI から GUI へ仕様変更することが行われる．従って，再利用の観点から UI に依存する部分と依存しない内部ロジックが分離されていることが望ましい．提案手法では，UI の変更としてアクティビティ図のユーザ/インタラクションパーティションの振る舞いと入出力データ構造の要素が変更される．一方，内部ロジック部分であるシステムパーティションの振る舞いと Entity データ構造は同一である．そして，Entity データとの関係を明記した入力データによる DTO クラスを用いることで，両者間の整合性を保つことができることから，変更は容易であると考えられる．

また，使用性の観点からテキスト入力をマウスによる入力方式に変更することが考えられる．マウス入力の場合は 2 点の座標が入力値となることから，Entity のデータ構造と対応させるための変換を行う必要がある．提案手法では，入力データの属性に Entity との派生関係を明記することで，変換内容を正しく実装まで落とし込むことができると考えられる．

5. 関連研究

Antović ら[2]はユースケースの基本/代替シナリオをドメインモデルの要素と指定されたアクションタイプを用いて記述し，ドメインモデルから UI 実現するテンプレートを決定することでテンプレートに基づいた Java GUI アプリケーションを自動生成する手法である．一方，本研究では，UI とシステムの内部ロジックを分離し，アプリケーションの本質と分離して UI の決定を行うために，両者間における整合性を保つ制約定義の関心事を設けることで，アプリケーションの本質から分離して UI の決定を吟味できることが特徴である．

6. まとめ

本稿では，UI とシステムの内部ロジックを分離したモデルを用いて機能を正しく実現するために必要な制約の定義を行うことにより，関心事を分離してモデルの整合性を管理しつつ，系統的に設計および実装を行うモデル駆動開発手法を提案した．また，仕様変更における関心事毎の依存関係について考察した．

今後は，より複雑な事例に適用することで手法の有効性を検討する．さらに，制約の定義を用いてモデルの整合性や妥当性を検査する方法の検討やユーザビリティの変化からなる UI と内部ロジックの影響を分析する．

参考文献

[1] 小形真平，松浦佐江子：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，コンピュータソフトウェア，vol.27，No.2，pp.14-32，2010

[2] I.Antović，et.al：Model and software tool for automatic generation of user interface based on use case and data model，IET Software，vol.6，iss.6，pp.559-573，2012

Copyright ©2015 Information Processing Society of Japan.
All Rights Reserved.