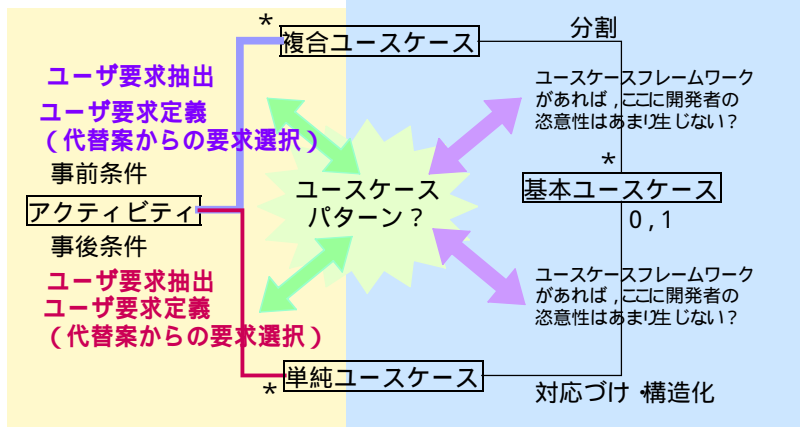


要求モデル



ユースケースフレームワークの検討

ソフトウェアシンポジウム2000
2000年6月21-23日

中谷 多哉子 (SLagoon), 玉井 哲雄 (東京大学)

背景

- Jacobsonが定義したユースケースは,事前条件,事後条件,基本系列と代替系列から構成される
 - ユースケースの数の爆発や基本系列と代替系列の対応関係などを管理するには不十分
- 実際に定義されたユースケースには,多くのユースケースで同じ系列を持つものが存在する
 - 記述の生産性を向上させるためには,同様の記述を再利用する枠組みが必要
- UMLのユースケースのメタモデルは現実的ではない
 - ユースケース間の関係が考慮されていない

目的

- ユースケースの明確な構造を定義する
- ユースケースの記述自体を統一するための記述の枠組みの提供
- 効果
 - ユースケースの記述の生産性向上
 - レビューの効率向上

関連研究1 : Jacobson

- Jacobsonによるユースケースの分類
 - 抽象ユースケース
 - ▶ 他のユースケースに拡張される
 - 具象ユースケース
- 構造
 - 名前 , アクター , 事前 / 事後条件 , 基本系列 , 代替系列

関連研究2 : 優先順位付け

- 基本系列と代替系列の改善
 - プライマリ
 - セカンダリ
 - オプショナル
- どの開発の優先順位付けを付加
 - Larman
 - SchneiderとWinters

関連研究3 : Cockburのユースケーステンプレート

- ゴール
- スcope : システム設計で考慮するブラックボックスの範囲
- レベル : Summary, Primary task, Subfunction
- 事前条件
- 成功時の事後条件
- 例外時の事後条件
- プライマリアクター a role name, or description
- トリガー
- 基本系列 (Main Success Scenario)
 - <step #><action description>
- 代替系列 (Extensions)
 - <step altered><condition>:<action or sub.usecase>

Cockburn続き

- 多様性
 - <step or variation #><list of sub-vcariations>
- 関連情報
 - 優先順位 , 処理効率の目標 , 利用頻度 , 当該ユースケースを(から)参照する(される)ユースケース , プライマリアクターのチャンネル , セカンダリアクター , セカンダリアクターのチャンネル
- 検討事項 (Open Issue)
- スケジュール

関連研究4：吉田ら

- ・ ユースケースの名前と管理用ID
- ・ 概要(Summary)
- ・ アクター(Actors)
- ・ 前提条件(Preconditions)
- ・ 処理(Descriptions)
- ・ 例外事項(Exceptions)
 - 入力の誤り,入力データ間の矛盾,オブジェクトの属性値自体の問題,属性値間の矛盾,入力データと属性値間の矛盾に起因する事項
- ・ 後条件(Postconditions)

Jacobsonの構造を持つユースケース例

- ・ ユースケース名: プログラム委員登録ユースケース
- ・ 事前条件: アクターがプログラム委員を登録するために必要な情報を得ている
- ・ 事後条件: プログラム委員が登録済みとなっている
- ・ 基本系列:
 1. アクターはシステムにプログラム委員を登録する旨を通知する。
 2. システムはアクターにプログラム委員を登録するための情報を要求する。
 3. アクターはシステムにプログラム委員の登録に必要な情報を渡す。
 4. システムは渡されたデータが正当である条件を満たし,かつ既に同じデータが登録されていないことを確認し,プログラム委員を登録する。
 - ・ プログラム委員登録に必要な情報とその情報の制約は以下のとおりである。
 - ・ プログラム委員名(文字列),所属(文字列),役職(文字列)*,連絡先(文字列),専門分野(文字列),関係国(文字列)(*はオプション)。ただし,関係国はすでに登録されている国名とする。また,重複して同じ人を登録しない。データの同一性は,名前と連絡先によって判断する。

改善への着目点

- ・ ユースケース名: **プログラム委員**登録ユースケース
- ・ 事前条件: アクターが**プログラム委員**を登録するために必要な情報を得ている
- ・ 事後条件: **プログラム委員**が登録済みとなっている
- ・ 基本系列:
 1. アクターはシステムに**プログラム委員**を登録する旨を通知する。
 2. システムはアクターに**プログラム委員**を登録するための情報を要求する。
 3. アクターはシステムに**プログラム委員**の登録に必要な情報を渡す。
 4. システムは渡されたデータが正当である条件を満たし,かつ既に同じデータが登録されていないことを確認し,**プログラム委員**を登録する。
 - ・ **プログラム委員**登録に必要な情報とその情報の制約は以下のとおりである。
 - ・ **プログラム委員名**(文字列),**所属**(文字列),**役職**(文字列)*,**連絡先**(文字列),**専門分野**(文字列),**関係国**(文字列)(*はオプション)。ただし,関係国はすでに登録されている国名とする。また,重複して同じ人を登録しない。データの同一性は,名前と連絡先によって判断する。

代替系列の構造化の検討

1. アクターが登録希望を取りやめたときは,**プログラム委員**の登録を中止し,副作用を残さない
2. **基本系列4**.において,正当なデータと判定できなかった場合は次のことを行う。
 - システムはデータが不当である理由を明らかにし,システムはアクターにデータの不当箇所と不当理由を通知する。
 - **基本系列2-4**を繰り返す。
3. **基本系列4**.において,登録しようとした**プログラム委員**が既に登録済みとなっている場合は次のことを行う。
 - システムは,アクターに登録希望された**プログラム委員**が登録済みとなっていることを通知する。
 - **基本系列2-4**を繰り返す。

ユースケースの特徴

- 同様の目的を達成するために記述されたユースケースにおいて、同様の事前条件、事後条件が定義される傾向がある。
- アクターとシステムとの相互作用は、同様の目的であれば、同様の順番で定義される傾向がある。
- 特定の基本系列項目に対して、同様の代替系列が定義される傾向がある。
- 特定の処理に対しては同様の利用者インターフェースが開発される傾向がある。
- 複数のユースケースの処理対象が同じオブジェクトである場合、オブジェクトに関する記述が繰り返し説明される傾向がある。

ユースケースフレームワークの例1

- ユースケース名：{登録対象オブジェクト}登録ユースケース
- アクター：{オブジェクト登録希望者}
- 目的：登録対象オブジェクトを登録する
- 事前条件：アクターが{登録に必要な情報}を得ている
- 正常終了における事後条件：
 - 登録対象オブジェクトが登録済みとなったという結果をアクターが得ている
- 例外終了における事後条件：
 - 登録対象オブジェクトの登録が中止され、副作用が残っていない
- ユースケース起動動作：
 - アクターはシステムに登録対象オブジェクトの登録希望を{サービス要求インターフェース}を介して通知する
- ユースケース全体にかかる代替系列：
 - アクターが登録希望を取りやめたときは、登録対象オブジェクトの登録を中止し、副作用を残さない

ユースケースの構造

- システム名：
- ユースケース種別：
- ユースケース名：
- アクター：
- 目的：
- 事前条件：
- 正常終了における事後条件：
- 例外終了における事後条件：
- ユースケース起動動作：
- ユースケース全体にかかる代替系列：
 - 基本系列：
 - 基本系列1. 動作説明
 - 基本系列2. 動作説明
 - ……
 - 基本系列n. 動作説明

各項目に多様性項目を定義し、同様のユースケースの記述の重複を解消する

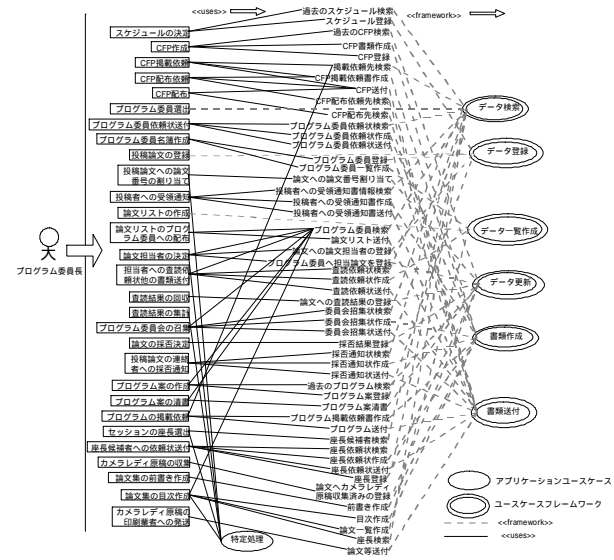
基本系列：

1. システムはアクターに登録対象オブジェクトを登録するための{登録インターフェース}を提示する。
 - …代替系列1-1: 登録インターフェースが提示されない
2. アクターが入手した登録インターフェースを介してシステムに登録対象オブジェクトの登録に必要な情報を渡すと、システムは渡されたデータが{データ制約}を満たし、かつ既に同じデータが登録されていないことを確認し、登録対象オブジェクトを{永続化}する。
 - …代替系列2-1: データが正当である条件を満たしていなかった場合は次のことを行う。
 - …2-1-1. システムはデータが不当である理由を明らかにする。
 - …2-1-2. システムはアクターにデータの不当箇所と不当理由を{データ不当箇所 / 不当理由通知インターフェース}を介して通知する。
 - …2-1-3. 基本系列2を繰り返す。
 - …代替系列2-2: 登録しようとした登録対象オブジェクトが既に登録済みとなっている場合は、次のことを行う。
 - …2-2-1. システムは、アクターに登録希望された登録対象オブジェクトが登録済みとなっていることを{重複登録不許可通知インターフェース}を介して通知する。
 - …2-2-2. 基本系列2を繰り返す。
3. システムはアクターに登録対象オブジェクトが新たに登録されたことを(登録対象オブジェクト登録完了通知インターフェース)を介して通知する。
 - …代替系列3-1: {通知に失敗した}
4. アクターは、登録対象オブジェクトが登録されたことを確認する。
 - …代替系列4-1: {確認に失敗した}

評価

- データの登録 / 検索 / 更新 / 一覧表作成 / 書類作成 / 書類送付 という6種類のユースケースフレームワークを定義
- 情報処理学会ソフトウェア工学研究会の要求工学ワーキンググループで採用されている共通例題「プログラム委員長業務」のユースケースを記述

複合ユースケース数	29
基本ユースケース数	58
アプリケーションユースケース数	53
ユースケースフレームワーク数	6



記述例

- システム名=プログラム委員長業務
- ユースケース種別=登録ユースケースフレームワーク
- 登録対象オブジェクト=プログラム委員
- オブジェクト登録希望者=プログラム委員長
- 登録に必要な情報=プログラム委員名(文字列), 所属(文字列), 役職(文字列)
*, 連絡先(文字列), 専門分野(文字列), 関係国(文字列) (*はオプション)
- サービス要求インターフェース=未定
- 登録インターフェース=未定
- ::代替系列1-1: 登録インターフェースが提示されない=考慮しない
- データ制約= 関係国はすでに登録されている国名とする。また、重複して同じ人を登録しない。データの同一性は、名前と連絡先によって判断すること
- 永続化=XXデータベースへ登録する
- データ不当箇所 / 不当理由通知インターフェース=未定
- 重複登録不許可通知インターフェース=未定
- 登録対象オブジェクト登録完了通知インターフェース=未定
- :::代替系列3-1: 通知に失敗した=考慮しない
- 登録対象オブジェクト登録確認インターフェース=未定
- :::代替系列4-1: 確認に失敗した=考慮しない

考察と結論

- アプリケーションユースケースの構造を統一できる。
- 基本系列の定型的な流れをフレームワークに定義できるため、システムの動作に一貫性を持たせることが容易となる。
- ユースケースフレームワークを介して、特定のユースケースでは、基本系列でどのような代替系列が発生し得るかといった熟練者のノウハウを提供することができる。
- アプリケーションユースケースで重複する部分を記述する必要がなくなり、ユースケースの記述の生産性が向上する。
- 個々のユースケースの差分は、明示的に示された多様性項目によって明記され、関係者によって重点的にレビューされるべき箇所の可視性が高まる。

まとめ

- ・ これまでに提案されたユースケースの構造を整理し、ユースケースフレームワークを提案した。
- ・ ユースケースフレームワークを利用することによって、ユースケースの生産性向上とレビューの効率向上も実現できる。
- ・ 今後は、ユースケースメタモデルによる記述支援とユースケースの管理を行い、ユースケースの品質向上を検討する。