

UML を用いた決定表の自動作成方式の提案

3Q-01

細川 卓誠 岡本 鉄兵 小泉 寿男
東京電機大学 理工学部

1. はじめに

オブジェクト指向は UML を用いて対象システムをモデル化し、顧客や技術者間で効果的に対話することを可能にする。顧客との対話で窓口的な役割を果たすのがユースケースである。ユースケースはユースケース図とユースケース記述からなり、お互いに補完し合い顧客の要求を表現している。しかし、これらだけでは全ての要求を表現しきれない[1]。例えば、組込みシステムで重要な要素である起動順序、平行実行可能性等の時間的制約などである。これらはユースケースの表すモデルでは表現する事が出来ない。本稿ではユースケースの不足する情報を補完する手段として、表思考である決定表に注目し、UML から自動作成する方式を提案する。

2. 決定表の自動作成方式

2.1 本方式の概要

方式全体の流れを図 1 に示す。(①)一般的にユースケースへのインプットは不完全な要求であるため、作成されるユースケースも不完全なものとなる。(②)しかし、最初から全てを記述する必要はないので、そのまま次の工程へと進む。(③)データベース内にある開発事例を基にクラス図を描く。(④)その時に変更すべき箇所がある場合は変更を行い、ユースケースへフィードバックして変更結果を反映させる。変更結果を反映したユースケースは一步洗練されたユースケースとなる。(⑤)新たに変更されたユースケースを基にアクティビティ図を描き変更を許すかどうか検討し、顧客の同意を得る。問題がある場合は工程②に戻り、一連の動作を繰り返しながら、要求定義を行う。(⑥)問題がない場合は工程⑥へ進み、最終的な結果を変換アルゴリズムに通して決定表を得る。

また、本方式を適用するにあたって、過去の開発事例が再利用可能な状態でデータベース内に蓄積されている必要がある。データベース内に蓄えられているデータは、クラス図、状態遷移図やアクティビティ図のドメイン別パターンや、用語辞書など開発に関

するさまざまな情報である。

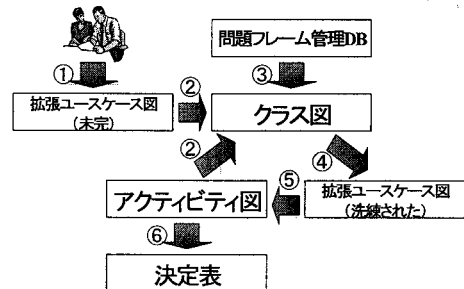


図 1 方式全体の流れ

2.2 本方式内で使用するモデル

本方式で使用する UML モデルはクラス図とアクティビティ図である。さらに本稿で提案する状態遷移の要素を付加した拡張ユースケースを用いる。なお、クラス図とアクティビティ図は UMLver1.3 に準拠したものである。

(1) 拡張ユースケース

ユースケース(オブジェクト)の性質で階層ごとに分けて記述する。拡張ユースケースを構成する追加削除可能な構成子(オブジェクト)として use case, actor の 2 つを用意する。さらに use case には境界、制御、実体といった 3 つの性質を持ち、それぞれに事前/事後条件、遷移条件などの情報を有している。また、前述の 2 つのオブジェクトは、それぞれにステレオタイプとして <<include>>, <<extend>> することが出来る。

(2) クラス図

クラス図はクラスとオブジェクトの観点からシステムの構造を示し、システムをオブジェクト指向で分解する上で、主要なガイドラインとなる。

(3) アクティビティ図

クラス図のフィードバックを繰り返すといったドメインの静的な構造を中心に考えていたため、アクティビティ図を描くことでドメインオブジェクトの振る舞いを考える必要がある。このことは、決定表作成の前ステップでもあるが、思考の偏りを訂正するという意味もある。これにより、ドメインモデルの誤りを早く発見する事も期待できる。

2.3 各ダイアグラム間での対応関係

本方式内では、各ダイアグラム間での対応関係を以下のようにしている。

拡張ユースケース	クラス図	アクティビティ図
ユースケース	操作(ヤシグナル)	アクティビティ
アクター	オブジェクトクラス	レーン
事前/事後条件	データ構造(振る舞い)	遷移の条件
データ	メッセージ	オブジェクト

図 2 各ダイアグラム間での対応関係

ただし、各ダイアグラムが等価であるということではなく、ベン図でいうところのANDの部分を抽出したに過ぎない。よって各ダイアグラムには対応しない部分も存在する。クラス図は必ずしも図2の通りに1対1で対応するとは限らない。また、クラス名、クラスの責任、他の協調クラスは表分割するパッケージ化の参考データになる。

2.4 決定表への変換アルゴリズム

決定表は、問題を処理するための条件と行動を表形式で表現することが可能である[2]。そのため、決定表は組込みシステムと相性がよく、ユースケース同様に特別な知識を必要としないといった可読性に優れる特徴がある。つまり読取側による誤解釈を招かないので、顧客との対話にも利用できる。

決定表は条件/処理記述部と条件/処理指定部の4つの部からなる。決定表の列(指定部)1つ1つを便宜上、処理決定規則と呼ぶことにする。

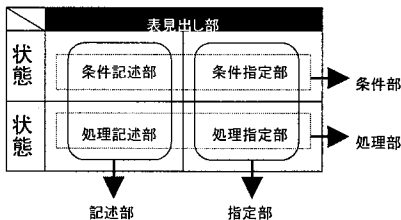


図 3 決定表の構造

アクティビティ図から2つのテーブル State Table, Transition Table を作成する。これらのテーブルはその名前通りの状態と遷移の情報を有する。テーブルのフォーマットを以下に示す。

- ・ State Table {name, do, entry, exit, event}
- ・ Transition Table {name, source, target, trigger, sync}

do,entry,exit は状態のアクティビティで,eventはその特有のインプットがある場合に動作する内容を記述する。source, target は事前/事後状態,trigger は遷移条件,sync は同期の有無をそれぞれ表す。これらをコンピュータへのインプットとしてデータ操作をし、決定表の自動作成を行う。条件部のデータとして

Transition Table を、処理部のデータとして State Table を用いる。また、記述部には関連アクターとしてアクティビティ図のレーンを用いる。処理決定規則は分岐オブジェクトのアウトブランチ数と通過経路を追跡することで決まる。

3. 適用例

以下に単機のエレベータモデルにおいて、ドア開閉動作の決定表の一部分を示す。通常、ドアはゴンドラとフロアの内外2つを考えるが、単純化するため、自動的に同期を取り開閉するものとした。エレベータの状態は昇降、停止、ドアの開中/閉中、開があり、関連する入力/出力は以下の通りである。

入力：開閉ボタン、開延長ボタン、戸閉安全装置、重さセンサ、タイマー

出力：スピーカ、戸の開閉

状態	条件	戸の開閉動作														
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
エレベータの停止	エレベータの停止	N	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
ドアの開中	完了遷移処理		N	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
	開閉ボタン押す		Y	-	-	-	N	N	N	N	N	N	N	N	N	N
	開延長ボタン押す		-	Y	-	-	N	N	N	N	N	N	N	N	N	N
	戸閉安全装置が働		-	-	Y	-	-	N	N	N	N	N	N	N	N	N
	重さセンサが働		-	-	-	Y	-	-	N	N	N	N	N	N	N	N
ドアの閉中	開閉ボタン押す						Y	-	-	-	-	-	-	-	-	-
	開延長ボタン押す						-	Y	-	-	-	-	-	-	-	-
	戸閉安全装置が働						-	-	Y	-	-	-	-	-	-	-
	重さセンサが働						-	-	-	Y	-	-	-	-	-	-
ドアの開中	完了遷移処理									N	Y					
	ドアが開							X	X	X	X			X	X	X
開	スピーカが音が鳴る										X					
	スピーカが音が鳴る							X								
動作終了	ドアが閉	X	X										X			

図 4 戸の開閉動作を表す決定表 (一部分)

4. まとめ

本稿では、決定表の自動作成方式を提案し、単機のエレベータモデルを用いて、結果を考察した。これにより、表として整理されている決定表は理解しやすいことが確認できた。また、図4より、処理の未定義な部分を発見できる場合もあることが分かった。本方式では、顧客が UML を知っている必要はなく、対象のシステムを把握することができる。しかし、アイコンの配置上の問題で、同等のオブジェクトが複数ある場合など、冗長な表になってしまうといった欠点もあった。

<参考文献>

- [1]吉田裕之,山本里枝子,上原忠弘,田中達雄,UMLによるオブジェクト指向開発実践ガイド.技術評論社,1999
- [2]T.G.Lewis 著,高橋延匡,松本正雄監訳,ソフトウェアエンジニアリング CASE の考え方と実践,日科技連,1993
- [4]渡辺 博之監修.リアルタイム UML 第2版.株式会社翔泳社.2001