

公理的設計法を用いたモジュール分割における横断的関心事の抽出

2003MT018 平崎 唯善

指導教員 沢田 篤史

1 はじめに

アスペクト指向技術によって、これまでのソフトウェア開発技術で解決し辛かった横断的関心事の分離が可能となった。横断的関心事とは、システムをモジュール分割した際に、複数のモジュールに散在する関心事のことである。分離が可能となっても、横断的関心事を開発の上流工程で特定することは困難である。上流工程における特定が望ましく、これは課題の一つに挙げられる。

本研究の目的は、横断的関心事の特定とモジュール化が可能な、システムの設計方法を提案することである。横断的関心事の特定が可能な設計方法を提案することで、アスペクト指向ソフトウェア開発を支援する。

本研究では、システムをオブジェクト、機能、非機能の視点から分析し、各視点の分析結果を対応付けることで横断的関心事の特定とモジュール化を行う方法を提案する。各視点の役割は、オブジェクトの抽出、機能の抽出、非機能の抽出である。そしてそれぞれに適した分析法による抽出を行うために、ユースケース、公理的設計法 [2]、NFR フレームワーク [1] を用いた。公理的設計法は機能の抽出と機能間の干渉を設計行列で表現することが可能な、機能視点のシステム設計法である。横断的関心事の候補は設計行列の干渉部分と、各視点の分析結果の対応関係の二点から特定できると考える。設計行列の干渉部分は他の機能と関わりを持つことから、横断的関心事の候補として扱うことができる。また、分析結果の対応関係からは、複数のオブジェクトに関わる機能や非機能を横断的関心事の候補と捉えることができる。

提案する方法の検証として、簡単な描画ソフトウェアの設計を行い、横断的関心事の特定を試みた。事例の結果から、提案する方法の妥当性について考察を行った。

事例を通して、複数の視点を用いた分析結果の対応付けによる横断的関心事の特定とモジュール化が可能であることが確認できた。このことから、横断的関心事の特定とモジュール化が可能な設計方法が提案できた。

2 関連研究と問題点

2.1 ADo-oSS

ADo-oSS(Axiomatic Design of Object-Oriented Software Systems)[3] は、公理的設計法を用いたオブジェクト指向ソフトウェアの設計である。機能要求とそれに対応する変数やクラス(設計解)を定義し、詳細化することで、システムの設計を行う。機能要求(以下,FR)と対になる設計解(以下,DP)を行列化して表現したものを設計行列(図1)と呼ぶが、この設計行列によって機能間の干渉を表現できる点が公理的設計法の特徴である。また、設計行列によって設計の評価を行うことが可能である。ADo-oSSでも設計行列を用いた設計を行う。

クラス名や変数名	DP1 ログ	DP2 部屋名	A,B: 機能要求と1:1対応する 設計解
FR1 ログを取る	A	0	X: 機能要求を満たすために 干渉する部分
FR2 予約する	X	B	0: 干渉がない部分

図1 設計行列の例

2.2 ADo-oSS の問題点

ADo-oSSの問題点として、オブジェクトの関係を表せないこと、余分な機能を持つオブジェクトが設計されてしまうことが挙げられる。これらの問題は、機能視点で設計した結果をオブジェクトに収めようとしたために起きた問題であると考えられる。また、全てのオブジェクトの関係を包含関係のみによって表現していたことも理由の一つに挙げられる。これらの問題から、本研究ではオブジェクトを考慮せず、機能の視点に限り公理的設計法を適用する。

3 提案方法の概要と公理的設計法の拡張

3.1 提案方法の概要

本研究では、オブジェクト、機能、非機能の三つの視点から設計を行う方法を提案する。提案する設計方法の概要を図2に示す。オブジェクトの抽出はユースケース、機能の抽出はADo-oSS、非機能の抽出はNFRフレームワークを用いて行う。それぞれの視点の設計を行い、結果を対応付けることでモジュールを構築する。

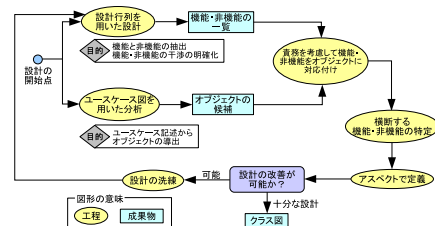


図2 提案する方法の概要

3.2 設計行列に現れる横断的関心事の候補

2.1節の図1に設計行列を示したが、この設計行列にはXという干渉部分が存在する。干渉部分は別のモジュールからの呼び出しや、条件判定が存在する場合に定義される。

上記から、干渉部分は横断的関心事になり得ると考える。干渉を含む機能や干渉部分自体が横断的関心事として妥当であるかを考慮して、干渉部分の分離を考える必要がある。

3.3 設計結果の対応付けによる横断的関心事の候補の特定

設計行列だけでなく、三つの視点の結果の対応関係からも横断的関心事の特定を行う。対応付けでは、それぞれの視点で分析した結果を責務を考慮して対応付ける。このとき、ある機能や非機能が、複数のオブジェクトで実装されなければならない場合、横断的関心事になり得ると考える。図3に対応付けの方法と横断的関心事と判

断できる機能の例を示す．この仮説を用いて，横断的関心事の特定を行う．

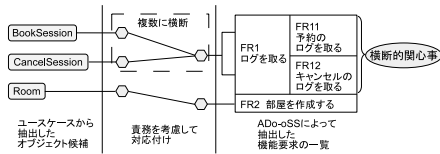


図3 対応付けと横断的関心事になり得る部分の例

3.4 NFR フレームワークの利用

公理的設計法では非機能要求の分析が行えないので，NFR フレームワークのカタログを利用して非機能要求の抽出を行う．カタログを用いた非機能の抽出結果は設計行列に適用する．機能との関係，各オブジェクトとの対応関係を明確にすることで，横断的関心事の候補を特定することができる．

4 事例検証と考察

4.1 事例の概要

事例検証として，簡単な描画ソフトウェアを用いる．描画ソフトウェアに対する要求を以下に示す．

- 線，円，長方形を描くことができる
- 各図形は異なる二点の座標を決定することで描画される
- マウスの操作によって動作する
- 描画領域外でリリースした場合，描画しない

4.2 設計行列から特定した横断的関心事の候補

機能，非機能視点の分析結果として，設計行列を作成する．設計行列の一部を抜粋したものを図4に示す．

		DP3 2個目の点			DP6 マウスイベント		DP7 登録するデータ	DPN1 料定箇所
		DP31 終点	DP32 円周点	DP33 右下頂点	DP61 フォックイベント	DP62 リリースイベント		
FR3	FR31 終点を得る	G						
FR3 2個目の点を得る	FR32 円周上の点を得る		H					
	FR33 右下の頂点を得る			I				
FR6 マウスに反応する	FR61 プッシュに反応する				O			
	FR62 リリースに反応する	X	X	X		P		
FR7	データを登録する	X	X	X			Q	
NFR1 例外処理	NFR11 リリース座標を判定する					X		U

FR: 機能要求(Functional Requirements)
 DP: 設計値(Design Parameters)
 NFR: 非機能要求(Non-Functional Requirements)
 DPN: 非機能用の設計値

X: 他の機能，非機能に干渉する部分
 横断する機能や非機能の明確化

図4 設計行列の一部

設計行列内の X は機能や非機能の干渉を示す部分であり，横断的関心事の候補になり得ると考える．例として，FR7 の“データを登録する”という機能は，二個目の点を決定した後に呼び出される．この機能はメソッド呼び出しによって動作するので，別のクラスに呼び出しの記述が書かれる．よって，横断していると判断できる．

4.3 対応付けから特定した横断的関心事の候補

各視点の分析の結果を，責務を考慮して対応付ける．図5に，オブジェクトと機能の対応付けの例を示す．

FR11, 12, 13 は詳細な機能であるが，FR1 の“描画する”という関心事を詳細化したものと捉えることができる．このことから，“描画する”という関心事単位で，機能が実装されると考えられるオブジェクトに対応付けた．複数のオブジェクトに関わっていることから，FR1 は横断的関心事の候補であると判断できる．

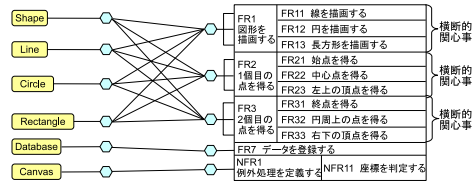


図5 オブジェクトと機能の対応付けの例

4.4 横断的関心事の候補の特定に関する考察

FR7 の“データを登録する”という機能は，対応付けでは機能とオブジェクトが一对一対応していることから，干渉部分が別の関心事を表していると考えた．FR7 に関する干渉部分を，“データベースへのアクセス”という関心事と捉え，分離するべきであると判断した．この関心事は対応付けからは特定できなかった関心事であり，干渉部分の意味を考慮する必要があった例である．設計行列と対応付けの二点から特定を行うことで，片方では特定できない部分を補うことができた．前述より，設計行列と対応付けの二点からの横断的関心事の特定は有効であると考えられる．

4.5 提案方法の妥当性の考察

描画ソフトウェアの設計では，横断的関心事の候補の仮説を用いて横断的関心事の候補を特定することができた．対応付けの結果と設計行列から，描画処理やデータベースへのアクセスといった機能的な横断的関心事を明らかにすることができた．今回の事例では提案する設計方法が有効だったと考える．

前述から，三つの視点の分析に異なる方法を用いて分析を行い，結果を対応付ける方法が有効であったと考える．ユースケースと NFR フレームワークを用いることで，公理的設計法のみで設計した場合の問題を解消し，分析結果の対応付けによって，複数のオブジェクトに横断する機能や非機能を明確にすることができたといえる．

5 おわりに

本研究では，横断的関心事の特定とモジュール化が可能な設計方法を提案した．提案する設計方法を用いた事例検証により，横断的関心事の特定に有効な設計方法であることが確認できた．

今後の課題としては，複数の事例で横断的関心事が特定可能かを考察する必要がある．さまざまな事例で横断的関心事を特定可能な設計方法であることを確認しなければならない．

参考文献

- [1] L. Chung, B. A. Nixon, E. Yu and J. Mylopoulos, *Non-Functional Requirements in software engineering*, Kluwer Academic Publishers, 2000.
- [2] N. P. Suh, *Axiomatic Design: Advances and Applications*, Oxford University Press, 2001.
- [3] S. -H. DO and N. P. Suh, “Object-Oriented Software Design With Axiomatic Design,” in *Proc. ICAD2000*, June. 2000, pp. 278-284.