

# アスペクト指向アーキテクチャ設計のための横断的関心事の抽出に関する研究

— アーキテクチャスタイルを用いて —

2007MI202 佐々木 研二      2007MI255 若尾 明日香      2007MI263 山下 優理

指導教員 野呂 昌満      沢田 篤史      蜂巢 吉成

## 1 はじめに

システム開発において、アーキテクチャは重要な役割を果たす [8]。また、システムを多角的に捉えて横断的関心事を分離する設計を行う必要があり、アスペクト指向技術が研究されてきた [2]。システムを設計する際、アーキテクチャと横断的関心事を共に考慮することは困難である。関心事からアーキテクチャを決定できず、またアーキテクチャから横断的関心事を識別することができないからである。我々はこの問題に対して、アーキテクチャと横断的関心事の関係を整理することで解決できると考えた。

本研究の目的は、アーキテクチャと横断的関心事の関係を整理することで、横断的関心事を考慮したアーキテクチャ設計が行えるようにすることである。

アーキテクチャは複数の視点から捉える必要があり、特に静的構造、動的振舞いの視点が重要である。これは Views and Beyond[5]にも示されている。我々は横断的関心事とアーキテクチャの関係を整理するために、静的構造、動的振舞い、横断的関心事の関係を整理した。これら3つの関係の整理には GoF のデザインパターン [1]を用いた。本研究の目的はシステムレベルのアーキテクチャと横断的関心事の関係を整理であり、デザインパターンは横断的関心事を実現するための静的構造、動的振舞いをコードレベルで示していることから、3つの関係を整理するのに適している。動的振舞いは Component-and-Connector Viewtype を用いる。Component-and-Connector Viewtype のスタイルでデザインパターンを分類し、分類したスタイルの静的構造を Module Viewtype の視点で抽象化することで、構造を新たに定義する。この静的構造をコンポジットスタイルと呼ぶ。コンポジットスタイルが表す関心事は、基となったデザインパターンが表す関心事である。デザインパターンが示す関心事を ISO9126[3]の品質特性で分類する。ISO9126は非機能要求について整理したものであり、一般的に横断的関心事は非機能要求であることが多い。このアプローチから、我々は横断的関心事とアーキテクチャの対応カタログを作成した。このカタログの妥当性は、Shaw と Garlan が提唱するアーキテクチャスタイル [4]と比較することで確認した。Shaw と Garlan が提唱するアーキテクチャスタイルは、静的構造、動的振舞い、横断的関心事について整理されており、一般的に受け入れられたものである。また、我々が作成したカタログを用いて横断的関心事を考慮したアーキテクチャ

設計を行えるかどうかを、飛行船自動航行システムの開発を通して検証した。設計は我々の研究室で提案する組込みシステムのためのアスペクト指向ソフトウェアアーキテクチャスタイル (E-AoSAS++ [7]) に基づいた。結果として、カタログを用いて飛行船のアーキテクチャを設計することができた。

成果として、アーキテクチャと横断的関心事の関係を整理したカタログを用いることで、アーキテクチャから横断的関心事を識別すること、また横断的関心事からアーキテクチャを決定することが容易となった。

## 2 アーキテクチャ設計と関心事

### 2.1 アスペクト指向アーキテクチャ設計における問題点

アスペクト指向設計とは、互いに関連する複数の関心事を分離して設計する方法であり、これを用いることで複数視点からのソフトウェア構造とそれらの間の関係を表現することができる。アスペクト指向アーキテクチャ設計における問題点として、アーキテクチャと横断的関心事の関係は必ずしも明確になっていないことが挙げられる。横断的関心事からどのようなアーキテクチャになるかを決定することができず、また逆にアーキテクチャの形からどのような横断的関心事が存在するかを識別することもできない。それゆえ、横断的関心事の抽出は容易でなく、アーキテクチャ設計が困難である。

### 2.2 アーキテクチャの設計視点

本研究では、アーキテクチャと関心事を考察する際にアーキテクチャの静的視点と動的視点の整理に Views and Beyond を用いた。Views and Beyond はシステムを捉える視点とその記述方式について整理されている。Views and Beyond ではシステムを3つの視点で捉えており、システムがこの3つの視点を満たしていれば十分であると言える。その3つの視点は、実装部位をどのように組み立てるかを表した Module Viewtype、実行時の振舞いと相互作用のある要素をどのように組み立てるかを表す Component-and-Connector Viewtype、環境においてソフトウェアでない構成物とどのように関連するかを表した Allocation Viewtype である。各 Viewtype にはそれぞれの Viewtype の範囲で頻繁に表れる表現を抽出し、一般化したものを style として定義している。

### 2.3 横断的関心事と非機能要求

一般的に横断的関心事となるのは非機能要求であることが多い。本研究では静的構造と動的振舞いを整理した結果明らかになった横断的関心事を ISO9126 を用いて

分類した。ISO9126 はソフトウェア品質の評価に関する国際基準である。ISO9126 はソフトウェアの持つ様々な特徴を品質の観点から整理したものであり、機能性、保守性などの 6 つの「品質特性」と品質特性をより細かく分類した 27 つの「品質副特性」とから構成されている。ISO9126 は完成したソフトウェアの品質の評価だけでなく、開発前のソフトウェアの機能要求や非機能要求を定義することも目的としている。この ISO9126 の分類を用いてソフトウェアに要求されている非機能要求を照合することで、非機能要求を網羅的に確認できるようになっている。

#### 2.4 関心事とアーキテクチャの関連付け

本研究で関心事とアーキテクチャの関連付けに GoF のデザインパターンを用いた。デザインパターンはソフトウェアの設計のパターンカタログであり、デザインパターンは実現したい関心事からどのような静的構造、動的振舞いにするかが整理されている。

#### 2.5 アーキテクチャスタイル

本研究で我々が提案するコンポジットスタイルの妥当性の考察に、Shaw と Garlan の提唱したアーキテクチャスタイルを用いた。アーキテクチャスタイルは複数のアーキテクチャに共通してみられる設計原則を整理したものである。アーキテクチャスタイルはソフトウェアをどうモジュール分割するかという静的構造と、コンポーネントやコネクタタイプの表現様式やこれらの組合せに関する一連の制約といった動的振舞いの両方を定義している。また、各アーキテクチャスタイルにはそれを用いることによって実現できる関心事が存在する。

### 3 関心事とコンポジットスタイルの対応関係

Views and Beyond の Component and Connector Viewtype に属するスタイルの動的振舞いにより適応可能性の視点からデザインパターンを分類し、分類したデザインパターンの静的構造を Views and Beyond の Module Viewtype の視点で抽象化し、コンポジットスタイルを定義した。また、基となったデザインパターンが表す横断的関心事を品質特性により分類し、そのコンポジットスタイルが持つ横断的関心事としカタログ化した。

対応関係の考察の例として Pipe-and-Filter コンポジットスタイルを挙げる。Component-and-Connector Viewtype の Pipe-and-Filter スタイルに分類したデザインパターンのうち、Composite パターン、Chain of Responsibility パターン、Strategy パターンに着目した。これらのパターンを用いることで Filter を切替えるという振舞いを実現できると考えた。アルゴリズムを切替える振舞いは、3 つのパターンの静的構造から、要求や命令が共通インターフェースを包含、または関連しており、共通インターフェースはアルゴリズムの抽象クラスとして汎化するという構造で実現できると考えた。以上より、定義した静的構造を図 1 に示す。これをコンポジットスタイルの静的構造とし、Pipe-and-Filter コン

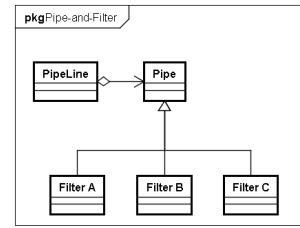


図 1 Pipe-and-Filter コンポジットスタイル

ポジットスタイルと名付けた。また、3 つのパターンの横断的関心事を調査し、品質特性で整理すると、資源効率性、変更性、障害許容性、安定性となる。これら 4 つの横断的関心事が、Pipe-and-Filter コンポジットスタイルの横断的関心事になると考えた。

同様に他の動的振舞いについてもコンポジットスタイルを定義し、その構造と関心事の対応関係について考察した結果を表 1 に示す。我々が作成した対応カタログには、コンポジットスタイルと品質特性の対応理由とコンポジットスタイルを用いることで損なわれる関心事が示されている。実際にアーキテクチャ設計を行う場合、1 つの関心事を選択すると複数のコンポジットスタイルが静的構造の候補として挙げられるが、コンポジットスタイルと品質特性の対応理由や、コンポジットスタイルを用いることで損なわれる関心事から、どのコンポジットスタイルを用いるかを決定する。

表 1 コンポジットスタイルと関心事の対応表

コンポジットスタイル	関心事
Pipe-and-Filter	変更性, 障害許容性, 安定性
PipeLine	変更性, 障害許容性, 安定性 セキュリティ, 解析性, 試験性
Shared-Data	変更性, 資源効率性, 安定性, 障害許容性
Visit-Shared-Data	変更性, 解析性, 時間効率性, 資源効率性
Iterator-Shared-Data	安定性, 時間効率性, 資源効率性, 環境適応性
Tree-Structure-Repository	変更性, 安定性, 解析性, 資源効率性
Publish-Subscribe	変更性, 安定性, 時間効率性
Medi-Client-Server	変更性, 安定性, 解析性, 時間効率性
Com-Client-Server	変更性, 障害許容性, 安定性 セキュリティ, 解析性, 試験性
Peer-to-Peer	変更性, 障害許容性, 解析性, 時間効率性
Communicating-Process	変更性, 安定性, 時間効率性
Decomposition-Process	安定性, 解析性, 試験性, 時間効率性

### 4 事例検証

MDD ロボットチャレンジ 2010[6] の仕様に基づいた飛行船自動航行システムを事例とし、コンポジットスタイルを用いてアーキテクチャ設計ができるかを検証した。

#### 4.1 コンポジットスタイルに基づくアーキテクチャ設計

飛行船自動航行システムにおいて満たすべき主要な要求を仕様書から分析し、横断的関心事を考察した。コンポジットスタイルを用いて、E-AoSAS++ の記述方法に基づき、飛行船自動航行システムの設計をした。

##### 4.1.1 関心事からのアーキテクチャ特定

表 2 に飛行船システムが実現すべき関心事を示す。

以下に、一例として Object-Oriented と Pipe-and-

表 2 関心事

関心事	コンポジットスタイル	使用した理由
変更性	Pipe-and-Filter	モータ出力計算アルゴリズムの変更性
時間効率性	Publish-Subscribe	飛行戦略の時間管理とセンス値を取得する間隔を管理する

Filter のアーキテクチャ設計の説明をする .

● Object-Oriented

我々は、飛行船自動航行システムではハードウェアを意識したモジュール分割が重要であると考え、支配的分割として Object-Oriented を適用した。クラス図を図 2 に示す。

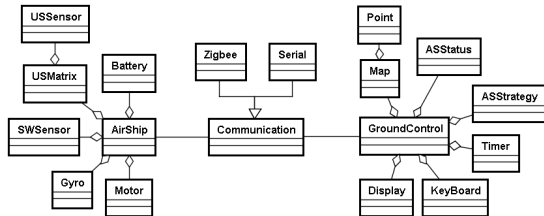


図 2 Object-Oriented で分割したクラス図

● Pipe-and-Filter

我々の飛行船自動航行システムの実現すべき関心事は、モータ出力値を演算するアルゴリズムを容易に追加、変更させることと、実行時に戦略に応じてアルゴリズムを切替える変更性が必要である。また、仕様からモータ出力演算を行うにあたってデータ構造の変更は頻繁には無いことがわかった。以上を考慮して、対応カタログから複数ある変更性を実現できるコンポジットスタイルの候補から対応理由を参考に Pipe-and-Filter が適切であると考え適用した。図 3 にその図を示す。図 4 には Object-Oriented と Pipe-and-Filter を用いて E-AoSAS++ に基づくシステムの静的構造を示す。

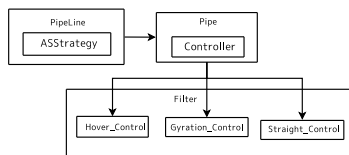


図 3 Pipe-and-Filter の動的振舞い

Pipe-and-Filter コンポジットスタイルを用いることで Control 部を分割することができ、飛行戦略部とモータ出力アルゴリズムの変更性を持たせる事ができた。その結果、我々が他の研究室に委託している飛行船のモータ出力値を演算するアルゴリズム部分を他のモジュールから分離することができた。

4.1.2 アーキテクチャからの横断的関心事の特定

コンポジットスタイルを用いてアスペクト分割したアーキテクチャと、コンポジットスタイルの静的構造を

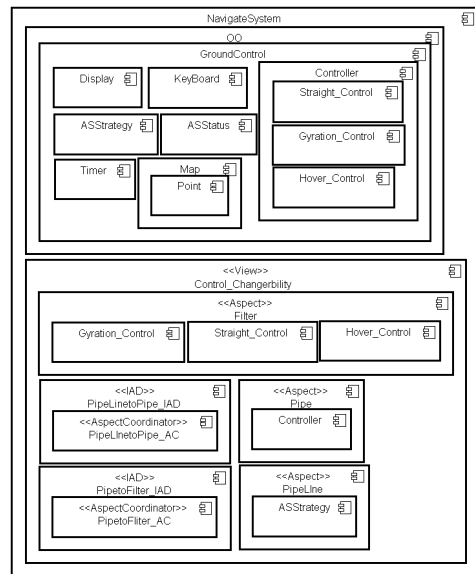
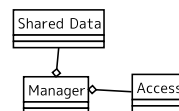


図 4 Pipe-and-Filter と Object-Oriented による静的構造

比較した結果、変更性、障害許容性、時間効率性以外にも、我々が要求分析した段階では発見できなかった横断的関心事があることがわかった。その一例として、Shared-Data の変更性について説明する。

飛行船自動航行システムにおいて飛行船の状態を格納しておく役割を持つ ASStatus クラスとその状態を演算し、ASStatus に格納するアルゴリズムを持つクラスがある。ASStatus クラスとアルゴリズムを持つクラスの関係がコンポジットスタイルの Shared Data の構造と一致した。したがって、データとデータに関する処理について変更性の関心事があることがわかった。図 5 にクラス図の構造とコンポジットスタイルの構造を示す。

Shared Data コンポジットスタイル



飛行船アーキテクチャ

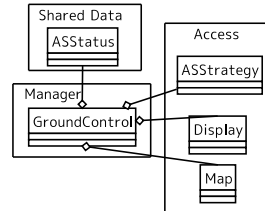


図 5 クラス構造とコンポジットスタイルの構造

5 考察

5.1 対応関係の妥当性の考察

我々が作成したカタログの、コンポジットスタイルと関心事の対応の妥当性を、コンポジットスタイルが表す横断的関心事と構造の対応、Shaw と Garlan が提

唱するアーキテクチャスタイルが表す横断的関心事と構造の対応の比較することで確認した。例として Shaw と Garlan が提唱するアーキテクチャスタイルの Pipe-and-Filter と、コンポジットスタイルを比較する。Pipes and Filters アーキテクチャスタイルの横断的関心事にはアルゴリズムや機能の変更が挙げられる。変更性をもつコンポジットスタイルは表 1 より Pipe-and-Filter, Shared-Data, Publish-Subscribe であるが、対応理由から Pipe-and-Filter コンポジットスタイルが最適であると判断する。Pipe-and-Filter コンポジットスタイルの静的構造は、図 1 に示す通り、Pipe が Filter を繋ぐという構造になっており、これはアーキテクチャスタイルの Pipes and Filters の構造と一致すると考えられる。以上のことから、Pipe-and-Filter の横断的関心事と構造の対応は妥当であるといえる。Shared-Data と Publish-Subscribe も同様に、構造と横断的関心事の対応が妥当であることを確認した。よって、コンポジットスタイルと関心事の対応が妥当であったと言える。

## 5.2 アプローチの妥当性に関する考察

5.1 節の考察から、コンポジットスタイルと関心事の対応が妥当であることがわかったが、コンポジットスタイルの動的振舞いから考えられる品質特性のうち、デザインパターンの横断的関心事に表れず、対応付かなかったものが存在する。例として、Publish-Subscribe の時間効率性を挙げる。Publish-Subscribe の動的振舞いから、1つのイベント発行で複数の Subscribe を操作でき、これは時間効率性に繋がるが、Views and Beyond の Public Subscribe に対応づいたデザインパターンからは導き出すことができなかった。このような対応関係を発見した場合は対応カタログに追加したが、対応していない横断的関心事がまだ存在することが考えられる。動的振舞いとデザインパターンの対応やデザインパターンと横断的関心事の対応など、コンポジットスタイルをより深く考察する必要があると言える。

## 5.3 横断的関心事の抽出に関する考察

本研究では、横断的関心事を考慮した設計が行えるようにするために、関心事と我々が提案したコンポジットスタイルとの対応関係を整理することで、アーキテクチャと横断的関心事の関係を整理した。我々は事例検証で、対応カタログを用いて、横断的関心事を考慮したアーキテクチャ設計ができるかを検証した。ここで、事例検証の Pipe-and-Filter コンポジットスタイルと、Shared-Data コンポジットスタイルを例に挙げる。飛行船を設計する際に要求として挙げた関心事の中に、飛行戦略とモータ出力演算アルゴリズムの追加、変更と実行時に戦略に応じてアルゴリズムを切替えるための変更性がある。また、仕様からデータ構造の変更は頻繁に行われなことがわかっている。対応カタログから飛行船の変更性を実現するアーキテクチャとして、Pipe-and-Filter コンポジットスタイルを用いた。その結果、Pipe-and-Filter を用いることで変更性を実現し、構造を決定することが可能であることを確認した。他の関心事

についても同様にアーキテクチャ設計を行うことができた。また、コンポジットスタイルを用いてモジュール分割を行った結果と Shared-Data コンポジットスタイルの静的構造を比較した結果、データに関する処理に関しての変更性が横断的関心事であることが確認できた。以上より、対応カタログを用いることで、横断的関心事を考慮したアーキテクチャ設計が可能となったと考える。

## 6 おわりに

本研究ではアスペクト指向アーキテクチャを設計するために、関心事とアーキテクチャの関係を整理し、関心事と我々の提案するコンポジットスタイルの対応カタログを作成した。対応カタログを用いて、関心事とアーキテクチャを考慮した設計法を提案した。関心事とコンポジットスタイルの対応関係を整理することで、関心事とアーキテクチャを共に考慮した設計を行えるようになった。また、コンポジットスタイルの動的振舞いから考えられる横断的関心事のうち、デザインパターンに表れず、対応付かなかった横断的関心事が存在した。対応していない横断的関心事がまだ存在することが考えられる。今後の課題として、動的振舞いの特徴やデザインパターンの構造の考察を続ける必要がある。

## 参考文献

- [1] E. Gamma, R. Helm, R. Johnson, and J. M. Vlissides, *Design Patterns: Elements of Reusable Object-Oriented Software*, Addison-Wesley, 1995.
- [2] F. Buschmann, R. Meunier, H. Rohnert, P. Sommerled, and M. Stal, *Pattern-Oriented Software Architecture: A System of Patterns*, Wiley, 1996.
- [3] ISO/IEC, *Software engineering Product quality - Part 1: Quality model*, 2001.
- [4] M. Shaw, D. Garlan, *Software architecture: perspectives on an emerging discipline*, Prentice-Hall, 1996.
- [5] P. Clements, F. Bachmann, L. Bass, D. Garkan, J. Ivers, R. Little, R. Nord, and J. Stafford, *Documenting Software Architectures Views and Beyond*, Addison Wesley, 2007.
- [6] MDD ロボットチャレンジ 2010 実行委員会, "MDD ロボットチャレンジ 2010 競技仕様書," [sdlab.sys.wakayama-u.ac.jp/mdd2010/MDD2010\\_regulation.pdf](http://sdlab.sys.wakayama-u.ac.jp/mdd2010/MDD2010_regulation.pdf), 2010.
- [7] 加藤大地, 蜂巣吉成, 沢田篤史, 野呂昌満, "アスペクト指向に基づくソフトウェアアーキテクチャの文書化方式," 信学技報 知能ソフトウェア工学研究会 (KBSE), Vol. 108, no. 449, pp. 55-60, 2009.
- [8] 野田勝彦, 呉曉星, 安竹由起夫, 岡村敦彦, 荒井玲子, "ラショナル統一プロセス ソフトウェア開発のベストプラクティス," UNISYS TECHNOLOGY REVIEW 第 68 号, MAR. 2001.