E-AoSAS++に基づく飛行船制御ソフトウェア開発
i*フレームワークを用いたソフトウェアアーキテクチャ設計

2006MI104 水野 紗央里　　2006MI140 太田 裕貴　　2006MI150 佐藤 綾香
指導教員　沢田 篤史　　蜂巣 吉成

1　はじめに

本研究室では保守性や再利用性の高いシステムを開発することを目的として，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 (以下，E-AoSAS++[4]) を提案している．E-AoSAS++ に基づく開発体系は PLSE[2] に基づいている．一般的に PLSE 開発に基づく研究において，要求仕様とアーキテクチャとの対応関係を考えることは，未解決な研究課題である．

本研究では，仕様モデルからアスペクトの候補を特定する方法を提案する．仕様モデルとアスペクト候補との対応関係を明確にすることで，系統的なアーキテクチャ設計を可能とし，省力化を目指す．

この目的を達成するために，本研究ではデザインパターンを手がかりに，要求分析結果と横断的関心事の対応関係を明確にし，カタログとしてまとめる．そして仕様モデルからアスペクト候補特定方法の仮説をたてる．このとき，要求分析にはゴール指向要求分析手法である i*フレームワークを用い，アーキテクチャには E-AoSAS++ に基づくアーキテクチャを用いた．

組込みシステムは外部環境やハードウェア制約などと密接に関係しており，機能要求だけでなく非機能要求を分析した仕様モデルを作成しなければならない．ゴール指向要求分析では非機能要求も分析することができる．また，i*フレームワークではシステムの利害関係者の視点ごとに要求を分析する．多視点から見た要求分析により，仕様モデル上には様々な関心事が表れる．その結果，横断的関心事を仕様モデル上から特定することが可能であると考え，i*フレームワークを採用した．

仕様モデルとアスペクト指向アーキテクチャとの対応関係を明らかにする手がかりとして，仕様モデルとデザインパターン [1] の対応関係を考えた．デザインパターンは，横断的関心事の種類 (意味) とアーキテクチャ構造の対応関係をまとめている．よって，仕様モデルから横断的関心事を特定し，アーキテクチャとの対応関係を明確にする手がかりになると考えたからである．

研究の手順は次の通りである．まず始めに i*フレームワークのモデルと，GoF デザインパターンとの対応関係を考えカタログ (以下，対応カタログ) を作成した．次に，形と意味から仕様モデルと対応づけた，支配的分割に横断する関心事が表れているモジュールが，アスペクト候補となるという仮説をたてた．そして実際に飛行船制御ソフトウェアを事例として，E-AoSAS++ に基づくアーキテクチャを設計し，仮説の検証，考察を行なった．

成果として，対応カタログを作成したことで，支配的分割であるオブジェクト指向に横断する関心事が明らかとなり，仕様モデルの形と意味からアスペクト候補の特定が容易になった．これにより，系統的なアーキテクチャ設計ならびにその省力化の基礎ができた．

2　E-AoSAS++

E-AoSAS++(Aspect-oriented Software Architecture Style for Embedded system) は，本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である．組込みソフトウェアにおける横断的関心事の問題を解決するために，E-AoSAS++ はアスペクト指向の考え方を導入している．E-AoSAS++ では，組込みソフトウェアを並行に動作する状態遷移機械 (以下，CSTM) の集合として規定しており，各 CSTM はイベントを受け取るとそのイベントに応じて状態が遷移する．そしてアクションとして各 CSTM の固有の処理を行う．各 CSTM が協調動作することで組込みソフトウェアの処理を実現する．

3　関連技術

3.1　ゴール指向要求分析

ゴール指向要求分析は，開発対象のシステムが達成すべき目的をゴールとし，サブゴールに段階的に詳細化することにより，システムの目標とその達成条件を明確化する要求分析手法である．

3.1.1　i*フレームワーク

i*フレームワークはゴール指向に基づいた要求分析手法の一つである．i*フレームワークは，アクタ，ゴール，タスク，ソフトゴール，資源という 5 つの要素を用いて，現状のビジネスを理解したり情報システム導入による効果などをモデル化し分析する手法である．i*フレームワークのモデルにはアクタ間の依存関係を分析する戦略依存 (Strategic Dependency: 以下，SD) モデルと，アクタ内部の依存関係を分析する戦略原理 (Strategic Rationale: 以下，SR) モデルの 2 つがある．i*フレームワークでは SD モデルと SR モデルを記述することによって，システムが何を求められているのかということと，求められていることを達成するためにどんな機能を持っているのかを分析することができる．

■ SD モデル

SD モデルでは、アクタ間の依存関係を表す．アクタとは、人，システム，役割などである．アクタは相手との関係に応じて受益者と提供者の立場が入れ替わる．i*フレームワークでは受益者をデペンダ (depender)，提供者

をデペンディ (dependee)，提供者が受益者から要求される何か (ソフトゴール，ゴール，タスク，資源) をデペンダム (dependum) という．SD モデルの構成要素を表 1 にまとめ, 記述例を図 1 に示す．

表 1　SD モデルの構成要素

名称	図形	説明
アクタ	アクタ名	人,システム,役割
ゴール	ゴール名	デペンダが所望する状態
タスク	タスク名	デペンダが実行する活動
ソフトゴール	ソフトゴール名	達成度の定量化が困難なデペンダが所望するゴール
資源	資源名	物理的実体,情報的実体
依存関係	デペンダ　デペンダム　デペンディ	デペンダムには,ソフトゴール,ゴール,タスク,資源を記述

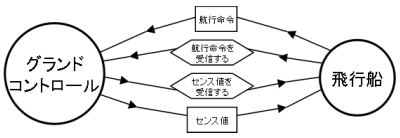

図 1　SD モデル記述例

■ SR モデル

SR モデルでは，アクタが何をどのように達成するのかについて段階的に詳細化する．タスク分解によってタスクをゴール，サブタスク，資源，ソフトゴールに分解することができる．同一の親タスクから分解された要素間の論理関係は AND 関係である．つまり子要素すべてを達成しなければ親タスクを実行することができない．手段目的分解では，4 種類の分解関係を記述することができる．同一の親要素から分解された要素間の論理関係は OR 関係である．したがって親要素を達成する代替手段を表現することができる．SR モデルの構成要素を表 2 にまとめ，記述例を図 2 に示す．

表 2　SR モデルの構成要素

関係	種類	要素の論理関係
タスク分解	◆タスク･ゴール関係 ◆タスク･サブタスク関係 ◆タスク･資源関係 ◆タスク･ソフトゴール関係	AND
手段目的関係	◆ゴール･タスク関係 ◆資源･タスク関係 ◆ゴール･サブゴール関係 ◆タスク･サブタスク関係	OR
貢献関係 +/−	ソフトゴールを目的とする分解関係 ◆ソフトゴール･タスク関係 ◆ソフトゴール･資源関係 ◆ソフトゴール･サブソフトゴール関係	＋:正の貢献 −:負の貢献

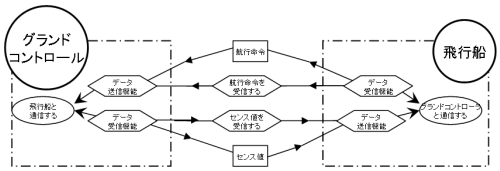

図 2　SR モデル記述例

3.2　GoF デザインパターン

GoF デザインパターンは，オブジェクト指向設計における特定の問題や課題を解決することができる 23 種類のパターンをカタログとしてまとめたものである．横断的関心事を実現するための，アーキテクチャ構造を定義しており，デザインパターンを適用することで，システムの保守性や再利用性などを向上することができる．

4　仕様モデルとアスペクト指向アーキテクチャの対応関係

仕様モデルとデザインパターンの対応関係を明らかにすることで，仕様モデルと，デザインパターンが実現する横断的関心事の対応関係を明らかにする．結果，いくつかの仕様モデルの記述パターン (以下，モデルパターン) には，複数のデザインパターンが対応することがわかった．以下の 11 種類のモデルパターンと, そのパターンに対応付いたデザインパターンをまとめて対応カタログとする．

表 3　対応カタログ

モデルパターン	対応するデザインパターン
連結型	Chain of Responsibility/Facade
選択型	State/Strategy
保守性向上	Builder/Factory Method Abstract Factory/Decorator Command/Bridge
機能性向上	Singleton/Abstract Factory Command/Proxy
効率性向上	Prototype/Singleton/Flyweight
部分選択型	Template Method
資源保存・復元	Mement
1 対多関連	Observer
多対多関連	Mediator
資源参照	Iterator
資源先制限	Singleton

例として State パターンと Strategy パターンを挙げる．この二つのパターンはともに場合によって委譲先を切替えるパターンであり，同じモデルパターン図 3 に対応付いた．このモデルパターンはタスク X の代替手

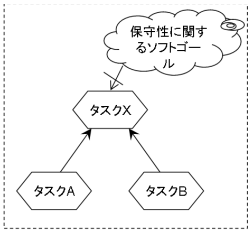

図 3　モデルパターン例：選択型モデルパターン

段としてタスク A と B が存在し，場合に応じてどちらか一方のタスクを実行することを表している．よって，このモデルパターンに『選択型モデルパターン』と名前をつけた．このモデルパターンが仕様モデル上に表れたとき，State パターンか Strategy パターンのどちらかが対応する可能性がある．どちらが対応するか判断するために，要素の意味に踏み込み次のように違いを明らかにした．

- State パターン：タスク A，B がモジュールの状態ごとの振舞い，ソフトゴールが振舞いの変更性

を表している場合

- Strategy パターン：タスク A，B が異なるアルゴリズムにより実現されている処理，ソフトゴールが実現アルゴリズムの変更性を表している場合

デザインパターンは支配的分割に横断する関心事を実現する．よって次のように仮説を立てた．

> 対応カタログにまとめた，横断的関心事と対応付けられた記述パターンが仕様モデル上に表れた場合，そのモジュールをアスペクト候補とする．

5 事例検証

事例を用いて，仮説によりアスペクト候補が仕様モデルから特定できるかの検証を行なう．事例には MDD ロボットチャレンジ 2009 の仕様 [3] に基づく自動航行飛行船制御ソフトウェアを用いた．

5.1 i*フレームワークによる要求分析

i*フレームワークは，要求の達成方法を分析することはできるが，どのような要求が存在するかは分析することはできない．ゴールを階層的にサブゴールへと詳細化することができないからである．これを補うために，われわれは i*フレームワークによる要求分析を行う前にゴールグラフを作成し，要求分析に必要となるゴールを抽出した．一部を図 4 に示す．そして，抽出したゴールをもとに SD モデルを作成した．SD モデルを作成するにあたり，われわれは MDD ロボットチャレンジ 2009 の仕様書に基づいたハードウェア分析を行なった．その結果として得た，システムを構成する 3 つの大きなハードウェアをアクタとする．さらにシステムと操作者の関係を明らかにするために，操作者をアクタに加え，アクタ間の依存関係の分析を行なった．次に，抽出したゴールと SD モデルをもとに，SR モデルを作成した．SR モデルを作成したことでアクタが何をどのように達成するのかが明らかとなった．

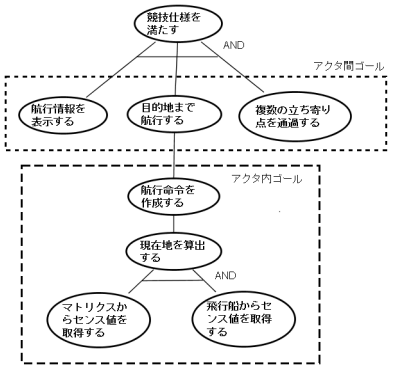

図 4　ゴールグラフ

5.2 モジュール分割

SD モデルと SR モデルから記述された要求分析結果から，クラスにモジュール分割を行う．われわれはクラスの責務にゴールが対応すると考え，ゴールを参考にクラスに分割した．この時，資源は情報的実体または物理的実体を表しており，本研究では分析対象がソフトウェアなのでデータにあたると考え，クラスの属性と考えた．タスクは属性である資源に対する処理を行なうので，操作と考えた．責務を表すゴール，属性を表す資源，操作を表すタスクをまとめてクラスとすることでオブジェクト指向に基づいたモジュール分割ができたと考える．

5.3 仮説の適用

対応カタログにまとめられたモデルパターンにあてはまる記述を，仕様モデル上から探した．そして仮説を用いてアスペクトの候補となるモジュールを特定する．仮説に基づきアスペクト候補としたクラスは以下の 5 つである．

- Navigation クラス
- Cache クラス
- Destination_Point クラス
- Analyzer クラス
- GC_Zigbee クラス

例として Navigation クラスを挙げる．Navigation クラスは図 5 のように仕様モデル上では表され，『選択型モデルパターン』の形に当てはまる．さらに要素が表す

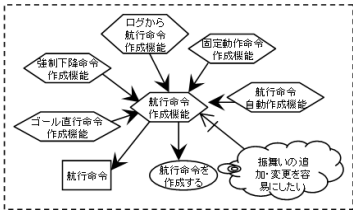

図 5　Navigation クラス

意味に踏み込み，対応するデザインパターンを特定する．Navigation クラスが持つソフトゴールは振舞いに関する変更性を表しており，各タスクは Navigation クラスの状態ごとに異なる航行命令の作成方法を表してる．よって Navigation クラスには State パターンが対応し，アスペクトの候補になった．

5.4 E-AoSAS++ に基づくアーキテクチャ設計

E-AoSAS++ に基づいてアーキテクチャ設計を行なった．5.3 節で仮説によって特定したクラスをアスペクトとし，設計したアーキテクチャを図 6 に示す．

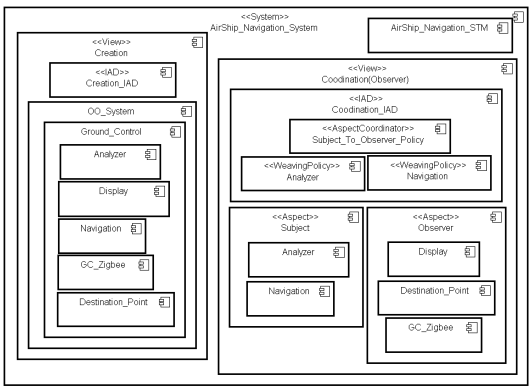

図 6　仮説を用いて設計したアスペクト指向アーキテクチャ (一部抜粋)

5.5 アスペクト候補の妥当性の検証

われわれは横断的関心事と対応しているモデルパターンの記述が仕様モデル上に表れたら，そのモジュールをアスペクト候補とする，という仮説をたてた．今回，対応を明らかにする手掛りとしてデザインパターンを用いた．仕様モデル上の記述とデザインパターンを対応付けることにより，デザインパターンが実現するオブジェクト指向に横断する関心事が，仕様モデルから特定できると考える．デザインパターンが実現する関心事は，品質特性に関する関心事である．

事例において，実際にアスペクト候補として特定されたクラスの例として，Analyzer クラスと Display クラスを挙げる．Analyzer クラスと Display クラスは 1 対多関連モデルパターンに対応している．よって，1 対多関連モデルパターンに対応付く Observer パターンが実現する，協調動作をするオブジェクト毎の保守性という関心事が，横断していると考える．支配的分割であるオブジェクト指向に他の関心事が横断しているので，Analyzer クラスと Display クラスはアスペクト候補として妥当と考える．他のアスペクト候補も同様である．

以上のことから，仕様モデル上から特定したアスペクト候補は妥当であると考える．

6 考察

6.1 仮説の有効性の考察

本研究で提案した仮説に基づき，別のドメインでも同様にアスペクト候補を特定することができれば，われわれの提案する仮説が有効であるといえる．

事例をプリンタにかえ，i*フレームワークによって要求を分析し，仕様モデルを作成してオブジェクト指向に基づきモジュール分割した (図 7)．この仕様モデルから，当てはまるデザインパターンがあるか，対応カタログを用いて探した．その結果，印字を責務としてもつ InkHeadding クラスが選択型モデルパターンの State パターンに対応していることが分かった．つまり，State パターンが実現する保守性に関する関心事が InkHeadding クラス上で横断している．仮説に基づきこのクラスをアスペクト候補とした．

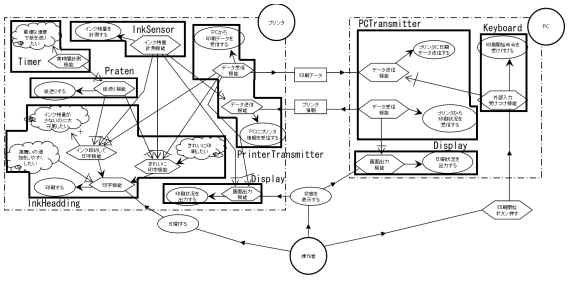

図 7　モジュールに分割したプリンターの仕様モデル

飛行船とプリンタは高い保守性を必要とするドメインである．また，本研究で作成した対応カタログで特定できる横断的関心事には，保守性に関する関心事が多い．よって，本研究で提案した仮説は有効であり，特に保守性に関する横断的関心事の特定にすぐれていると言える．

6.2 課題解決に関する考察

本研究では，デザインパターンが実現する関心事を整理し，仕様モデルの形と意味にデザインパターンを対応づけることで，要求分析の段階で横断的関心事を特定する仮説を提案した．これにより，仕様モデルと横断的関心事の対応関係が明確化したと考える．しかし，仕様モデルと対応付かないデザインパターンも存在した．例として Composite パターンを挙げる．Composite パターンは，i*フレームワークによる仕様モデルでは再帰的な構造を表せないことから，仕様モデル上の記述に対応付けることができなかった．よって，Composite パターンが実現する関心事である再帰的な構造を持ったオブジェクトの保守性が，支配的分割であるオブジェクト指向に横断していたとしても，仕様モデルから特定することはできない．この他，Adapter パターン，Interpreter パターン，Visitor パターンを対応付けることができなかった．

7 おわりに

本研究の目的は仕様モデルからアスペクト候補を特定し，仕様モデルとアスペクト指向アーキテクチャとの対応関係を明確にすることで，アーキテクチャ設計の省力化の支援をすることである．仕様モデルとデザインパターンとの対応関係を整理し，オブジェクト指向に横断する関心事をアスペクト候補として特定できるようになったことで，仕様モデルとアーキテクチャの静的側面との対応関係が明らかになった．

今後の課題として，対応付かなかったデザインパターンの実現する関心事が，組込みシステムにおいて仕様モデルから特定するべきものかを考察することが挙げられる．また，仕様モデルからこれらの関心事を特定する方法を提案することで，さらなるアーキテクチャ設計の支援が期待できる．

参考文献

[1] E. Gamma, R. Helm, R. Johnson, J. Vlissides, *Design Patterns*, Addison-Wesley Publishing Company, 1995.

[2] K.Pohl, G.Bockle, and F.Linden, *Software Product Line Enginnering: Foundations, Principles and Techniques*, Springer, 2005.

[3] MDD ロボットチャレンジ 2009 実行委員会，MDD ロボットチャレンジ 2009 競技仕様書，2009.

[4] 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.

[5] 山本修一郎，〜ゴール指向による!!〜システム要求管理技法，株式会社ソフトリサーチ，2007.