

# E-AoSAS++に基づく飛行船制御ソフトウェア開発 KAOSを用いたソフトウェアアーキテクチャ設計

2006MI152 佐藤 俊成

2006MI184 寺澤 将志

2006MI201 山田 洋平

指導教員 沢田 篤史

## 1 はじめに

近年、組み込みソフトウェアの規模や複雑さが増している。規模や複雑さの増加に対して、ソフトウェア開発における要求分析から実装に至るまでの追跡性を高めることが重要となっている。

本研究室では組み込みシステムのためのアスペクト指向アーキテクチャスタイルである E-AoSAS++[7] を提案している。アスペクト指向を適用することで、互いに関連する複数の視点によってソフトウェアをモジュール化し、再利用性や保守性を高めることができる。E-AoSAS++ に基づいた開発体系はプロダクトラインソフトウェアエンジニアリング (以下、PLSE[3]) の考えに基づいている。PLSE とは製品間で共通な部品をコア資産として再利用し、開発を行う開発方法論である。E-AoSAS++ を用いた開発を PLSE の方法論として見たとき、仕様モデルとアーキテクチャの対応関係が必ずしも明確ではないことが問題点として挙げられる。

本研究の目的は仕様モデルからアスペクト候補を特定することで、アーキテクチャとの対応関係を明確にすることである。仕様モデルからアーキテクチャの対応関係を考えるにあたり、ゴール指向分析手法である KAOS[2] を用いる。ゴール指向分析では機能要求だけでなく、非機能要求を分析することができるので、実時間処理や例外処理、耐故障性などの非機能要求に関する関心事が絡み合う組み込みソフトウェアの分析に適していると考えられる。また、KAOS は複数の視点からシステムを分析できるので、仕様モデル上にさまざまな関心事が表現でき、アスペクト候補を容易に特定できると考える。本研究では KAOS の分析に NFR フレームワーク [4] の考え方をを用いることで非機能要求を系統的に抽出する。

仕様モデルとアーキテクチャの対応関係を考えるにあたり、KAOS のゴールモデルに着目する。KAOS のゴールモデルはシステムの達成すべき目標状態 (以下、ゴール) をサブゴールに手段目的展開で段階的に詳細化していくことでユーザからの要求を達成するための手段を明確化する。KAOS のゴールモデルにおいて、ある一つの要件が複数のゴールを達成するために手段目的展開されたとき、アスペクト候補であると仮説をたて事例を用いて評価し、考察を行う。本研究では事例として、飛行船制御ソフトウェアを用いる。飛行船制御ソフトウェアは機能要求だけでなく、ハードウェア制約や外界からの影響を考慮した非機能要求の分析を行う必要があり、複数の視点からモジュール化を行うことが可能であると期待できる。

結果として、KAOS のゴールモデルの形と意味からアスペクト候補の特定を行うことが可能だと分かった。これらのことから、開発者はモジュール化を行う前にアスペクト候補を特定し、再利用可能な部品を開発、保守することができる。

## 2 アスペクト指向に基づくソフトウェアアーキテクチャ設計とその問題点

### 2.1 E-AoSAS++

E-AoSAS++ (Aspect-oriented Software Architecture Style for Embedded system) は、本研究室で提案する組み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。E-AoSAS++ では、組み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) の集合と規定している。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し、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み込みソフトウェアの処理を実現する。

### 2.2 アスペクト指向アーキテクチャ設計における問題点

E-AoSAS++ を用いた開発を PLSE の方法論として見たとき、仕様モデルとアーキテクチャの対応関係が必ずしも明確ではないことが問題点として挙げられる。アプリケーションドメインにおける概念構造をそのままアーキテクチャに反映させると類似する処理を持つモジュールが複数あられ、横断的関心事の問題が発生する。この問題を解決できないと PLSE の方法論に基づく開発を行うことはできないが、アスペクト指向を適用することで、横断的関心事の問題を解決することができる。しかし、仕様モデルから横断的関心事を系統的に抽出し、ソフトウェア全体をモジュール化する手法は確立されておらず、再利用可能なアスペクトとして部品を開発、保守することができない。本研究では、先に述べた問題を解決するために以下に挙げる手法を用いる。

### 2.3 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標をゴールとし、サブゴールに手段目的展開で段階的に詳細化することにより、達成手段を明確化する手法である。ゴール指向要求分析の特徴として、一般的な要求分析手法であるユースケースモデルでは扱えない非機能要求を分析することができることが挙げられる。また、複数の達成手段を抽出することができ、代替案の提案をすることができる点も特徴である。

## 2.4 KAOS

KAOSはゴール指向要求分析手法の一つであり、達成手段があいまいなゴールからシステム設計の選択肢を作りだすことができる。KAOSではゴールをAND/OR分割で、サブゴールに詳細化する。KAOSで扱う要素を図1に示す。

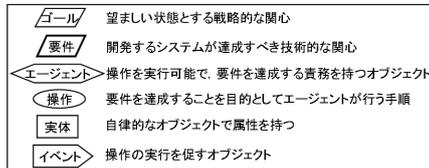


図1 KAOSの構成要素

KAOSに従う分析過程では異なる視点で分析したモデルとして、ゴールモデル、オブジェクトモデル、エージェント責任モデル、オペレーションモデルの4つのモデルを作成する。

### ● ゴールモデル

KAOSの分析の主導的モデルである。実現手段があいまいなゴールを開発すべきシステムが達成すべき要件になるまで手段目的展開を行う。また、各ゴールに対してゴール記述を記述し、ゴールを具体的に定義する。

### ● エージェント責任モデル

開発すべきシステムが達成すべき要件をエージェントに割り当てる。

### ● オブジェクトモデル

ゴールモデルから実体を抽出し、各実体同士の関係を定義する。アプリケーションドメインにおいて扱われているオブジェクト同士の関係を表す。

### ● オペレーションモデル

アプリケーションドメインにおいてエージェントが要件を達成するために行う操作を入出力、事前事後条件を用いて定義する。

## 2.5 NFRフレームワーク

NFRフレームワークは非機能要求の分析を主体とした分析手法である。NFRフレームワークでは、非機能要求をいくつかの種類にわけける分割方法をカタログとして定義してあり(以下、タイプカタログ)、分析者はタイプカタログを用いてドメインに必要な非機能要求を分析することができる。タイプカタログを用いることで非機能要求の系統的な抽出を可能にしている。さらにNFRフレームワークでは、非機能要求に対する実現手段を操作化カタログとして定義している。NFRフレームワークを用いることで、ドメインで考えられる非機能要求を分析し、その実現手段を明らかにすることができる。

## 3 KAOSを用いたソフトウェアアーキテクチャ設計

### 3.1 KAOSへのNFRフレームワークの考え方の導入

本研究では、KAOSの分析にNFRフレームワークの考え方を取り入れる手法を提案する。KAOSに従う

分析過程にNFRフレームワークの考え方を取り入れることによって、系統的に非機能要求を実現する手段が抽出され、開発者が見落としがちである非機能要求の分析支援を行う。KAOSのゴールモデルにおいて、ゴールがNFRフレームワークのタイプカタログの内容を示すとき、対応する操作化カタログの語彙を用いて要件を定義することで、機能要求と比べ特定が困難である非機能要求の実現手段を容易に決定することができる。NFRフレームワークの考え方をういたKAOSのゴールモデルにおいて、操作化カタログの語彙を用いて定義された要件は複数のゴールから手段目的展開された形で多数あらわれると考える。

### 3.2 アスペクト候補特定の仮説

KAOSのゴールモデルの形からアスペクト候補特定の仮説を立てる。KAOSのゴールモデルではゴールを手段目的展開を用いて、システムが達成すべき要件にまで展開する。手段目的展開で展開された上位ゴールを一つの関心事と捉えることで次の仮説を立てることができる。ある一つの要件が複数のゴールから手段目的展開されているとき、その要件は上位ゴールを関心事としたときの横断的関心事である。図2にアスペクト候補特定の仮説が適用されるゴールモデル例を示す。

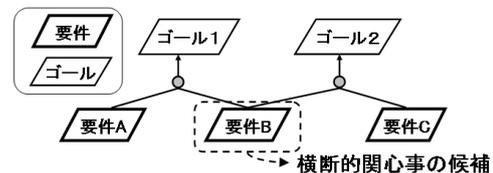


図2 アスペクト候補特定の仮説を適用するゴールモデル例

### 3.3 モジュール候補の特定

KAOSのオブジェクトモデルと、オペレーションモデルを用いて、モジュール候補の特定を行う。KAOSのオブジェクトモデルではアプリケーションドメインにおける実体、エージェントの関係を表しており、オペレーションモデルではアプリケーションドメインにおける要件を達成するためにエージェントが行う操作を表している。この二つのモデルを構成している要素の意味から、実体とそれに関係する操作を一つのモジュールとして扱えば、オブジェクト指向に基づくモジュール分割ができると考える。しかし、そのモジュール分割では横断的関心事の問題が解決されていない。我々は、KAOSのゴールモデルを用いて横断的関心事を特定する仮説をたてた。KAOSの各モデルは依存関係をもっており、操作がどの要件を達成するために抽出されたか追跡することが可能である。アスペクト候補特定の仮説で横断的関心事の候補となった要件から抽出された操作はアスペクトとしてモジュール化を行えば、横断的関心事の問題を解決することができる。したがって、我々が提案するKAOSのモデルを用いたモジュール候補の特定の手順は次のようになる。実体とそれに関係する操作を一つのモジュールとして扱うが、アスペクト候補として抽出された要件に関わる操作はアスペクトとしてモジュール

ル化する。

#### 4 事例による評価

MDD ロボットチャレンジ 2009[6] の仕様に基づいた飛行船制御ソフトウェアを事例として仮説の評価を行う。

##### 4.1 事例概要

事例として取り上げる飛行船制御ソフトウェアの概要は、MDD ロボットチャレンジの競技仕様書に記載されている。概要を以下に示す。

- 飛行船が出発地を離陸してから、いくつかの規定動作を行った後、立ち寄り点を通過しながら目的地に着陸する
- 規定動作は、基本となる動作を指定された順番で実行する
- 飛行船と地上に置かれたセンサと通信を行い、得られたセンサ値からモーターの制御値（以下、航行コマンド）を計算する
- 航行中は、航行状態をディスプレイに表示する

これらの概要を開発するソフトウェアが達成すべき目標状態としてとらえ、要求分析を行った。

##### 4.2 特定したアスペクト候補の評価

KAOS を用いた分析過程で作成したゴールモデルから、アスペクト候補特定の仮説が適用可能な箇所が複数箇所抽出された。図 3 にゴールモデル例を示す。

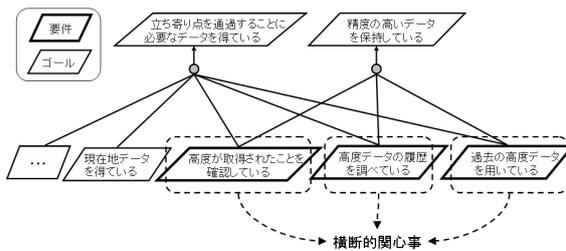


図 3 KAOS のゴールモデルの一部

“高度が取得されたことを確認している”という一つの要件は、“立ち寄り点を通過することに必要なデータを得ている”と“精度の高いデータを保持している”という2つのゴールに対して手段目的展開されている。“高度が取得されたことを確認している”という要件はアスペクト候補特定の仮説より横断的関心事である。“高度が取得されたことを確認している”から抽出される操作をモジュール分割の仮説に従い操作のみでモジュール化する。図 4 にモジュール分割の一例を示す。

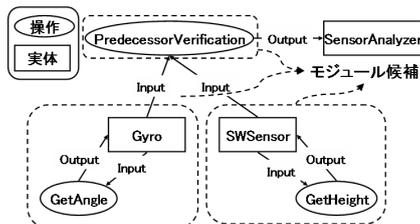


図 4 仮説に基づくモジュール分割例

実体“Gyro”は2つの操作“GetAngle”と“PredecessorVerification”と関係している。実体“SWSensor”は2つの操作“GetHeight”と“PredecessorVerification”と関係している。操作である“PredecessorVerification”はアスペクト候補特定の仮説とモジュール候補の特定から操作のみでモジュール化する。図 5 にゴールモデルの一部例を示す。

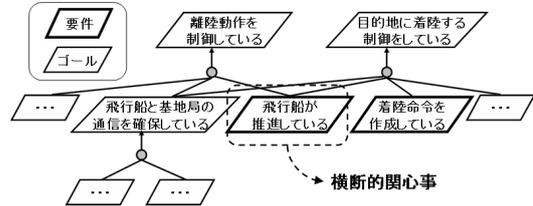


図 5 KAOS のゴールモデルの一部

要件の一つである“飛行船が推進している”は、飛行船の着陸動作を制御している”と“目的地に着陸する制御をしている”という複数のゴールに対して手段目的展開されている。アスペクト候補の仮説から“モーター出力をしている”は横断的関心事であり、アスペクトとしてモジュール化する。しかし、実体“Motor”からモーターを制御するための操作をアスペクトとして抽出することが妥当であるかは KAOS のゴールモデルの形からは判断することができないことがわかる。

#### 5 考察

##### 5.1 提案する仮説に関する考察

本研究で提案した“ある一つの要件が複数のゴールから手段目的展開されているとき、その要件は上位ゴールを関心事としたときの横断的関心事である”という仮説について考察する。

本研究では仕様モデルとアーキテクチャの対応関係を明確にするために、KAOS のゴールモデルの形からアスペクト候補の特定を行った。しかし、事例による評価でゴールモデルの形からのみでは、アスペクト候補の妥当性をいうことはできないと考えられる。ここで、横断的関心事として抽出された要件が表していることの意味にまで踏み込む。その要件がその他の要件に依存していれば、抽出したアスペクト候補は妥当であると言える。我々の提案する分析手法では、アスペクトとしてモジュール化した操作がどの要件から抽出されたか追跡することができ、横断的関心事として特定された要件がその他の関心事に依存していることを表すことができる。

##### 5.2 KAOS の分析での NFR フレームワークの考え方の貢献の考察

KAOS の非機能要求の分析に NFR フレームワークの考え方を取り入れることで、要件がどのようなソフトウェアの品質を向上させているのか追跡することが可能である。非機能要求は機能要求に依存しているから、ゴールモデルの形だけでなく、アスペクト候補特定の仮説から横断的関心事となった要件の意味から非機能要求だとわかり、アスペクトとして妥当であることがいえる。

よって、NFR フレームワークの考え方をを用いた KAOS の分析は仮説で特定された横断的関心事が妥当であるかどうか、追跡できる点に貢献がある。

### 5.3 作成したアーキテクチャに関する考察

KAOS を用いたモジュール分割で分割された各モジュールを CSTM とし、仕様モデルから特定したアスペクト候補がアーキテクチャ上に表現することができるか考察を行う。仕様モデルからアスペクト候補を系統的に特定することで、開発者に依存しない設計が可能となる。図 6 に作成した E-AoSAS++ のアーキテクチャ記述の一部を示す。

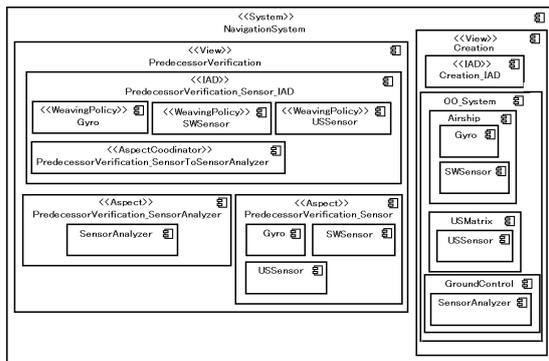


図 6 E-AoSAS++ のアーキテクチャ記述の一部

アスペクト候補特定の仮説から、導き出された要件である“データを取得していることを確認する”を横断的関心事として捉え、アスペクトとモジュール化する。“データを取得していることを確認する”に関する操作をアスペクトとしてモジュール化することで、操作と関係のあったモジュールの再利用性が向上する。

## 6 関連研究

Yu は、本研究と同様にゴール指向分析手法を用いて、アスペクト候補を特定する系統的な方法を提案している [5]。V グラフと呼ばれる独自のゴールモデルを用いて、機能要求と非機能要求に貢献するシステムが実行する機能 (以下、タスク) を表している。このとき複数の機能要求に関係しているタスクをアスペクト候補として特定している。機能要求と非機能要求を抽出する手順については言及しておらず、機能要求と非機能要求を系統的に抽出できる本研究で用いた分析が有効であると考えられる。

Baniassad らは、2 種類のテーマアプローチを用いてアスペクト候補を特定し、その後要求分析の結果を設計に反映させる方法を提案している [1]。要求分析段階で用いるモデルと設計段階で用いるモデルは 1 対 1 で対応することから、要求分析から設計までの追跡可能性が高まる。特定したアスペクト候補が妥当であるかについては言及されておらず、本研究で提案した分析手法と組み合わせることでアスペクト候補の意味から妥当性を示すことができると考える。

## 7 おわりに

本研究では、仕様モデルからアスペクト指向アーキテクチャを設計する際の問題点を挙げた。その問題点を解決するために、ゴール指向要求分析手法である KAOS を用いて、アスペクト候補を特定する仮説を立てた。次に飛行船制御ソフトウェアを事例として仕様モデルを作成し、仮説の評価と考察を行った。本研究の目的は仕様モデルとアーキテクチャとの対応関係を明確にすることであるが、仕様モデルからアスペクト候補特定の仮説を立てることで、横断的関心事の問題を解決し、開発者は再利用可能な部品を開発、保守することができる。今後の課題として、本研究で提案した手法ではゴールモデルの形と意味から横断的関心事となる要求を特定し、要件を達成するための操作をアスペクトとしてモジュール化した。このとき、アスペクト候補の仮説から特定されたモジュールが妥当でないものが存在した。この問題を解決するために、今後はより深くゴールの意味に踏み込む必要があると考えられる。

## 参考文献

- [1] E.Baniassad, and S.Clarke, “Theme: An Approach for Aspect-Oriented Analysis and Design,” Department of Computer Science, 2004.
- [2] E.Letier, *Reasoning about Agents in Goal-Oriented Requirements Engineering*, Universite Catholique de Louvain, Dept. Ingenierie Informatique, 2001.
- [3] K.Pohl, G.Bockle, and F.Linden, *Software product Line Engineering : Foundations, Principle and Techniques*, Springer, 2005.
- [4] L.Chung, B.A.Nixon, E.Yu, and J.Mylopoulos, *Non-Functional Requirements in Software Engineering*, Springer, 1999.
- [5] Y.Yu, “From Goals to Aspects: Discovering Aspects from Requirements Goal Models,” Julio Cesar Sampaio do Prado Leite, Department of Computer Science, 2004.
- [6] 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競技仕様書,” [http://sdlab.sys.wakayama-u.ac.jp/mdd2009/MDD2009\\_regulation.pdf](http://sdlab.sys.wakayama-u.ac.jp/mdd2009/MDD2009_regulation.pdf), 2009.
- [7] 加藤大地, 蜂巢吉成, 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, pp55-60, 2009.