

アスペクト指向アーキテクチャに基づく 組込みソフトウェア開発の支援 —自律走行ロボットのアーキテクチャの構築—

2010SE047 久田浩太郎 2010SE073 伊藤孝純

指導教員：沢田篤史

1 はじめに

近年、ユーザ要求の多様化によりシステムが大規模化し、その結果としてシステム内で横断的関心事が複雑に絡み合っている。それによりソフトウェアの保守性や再利用性の低下を招いている。横断的関心事をモジュール化して分離する技術としてアスペクト指向技術がある。アスペクト指向技術を用いて横断的関心事をアスペクトとして抽出することにより、ソフトウェアの保守性や再利用性を高めることができる。

本研究の目的は自律走行ロボットのアスペクト指向アーキテクチャの構築である。ESS ロボットチャレンジ 2013 で用いた自律走行ロボット Kobuki のアスペクト指向アーキテクチャを構築する。既存のアーキテクチャスタイルを組み合わせることでオブジェクト指向アーキテクチャを構築し、横断的関心事を整理することでアスペクト指向アーキテクチャを構築する。アーキテクチャスタイルは MVC とイベント駆動型アーキテクチャを組み合わせて用いる。横断的関心事は E-AoSAS++[2] で定義されている関心事と事例で現れると考えられる関心事に基づいて考察する。ESS ロボットチャレンジ 2013 の競技ルール [3] と Kobuki のハードウェアをもとに仕様を考察する。考察した仕様をモデル化することで必要な役割を明確にする。

仕様は Feature-Oriented Reuse Method(以下, FORM)[1] に基づくフィーチャモデルを用いてモデル化する。FORM とは構成要素であるフィーチャを 4 つの層に分類して木構造で記述することで構成要素の役割を明確にする手法である。E-AoSAS++ とは本研究室で提案されている組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。組込みシステムを並行して動作するひとつの状態遷移機械の集合と定義している。並行性、実時間性などの 5 つの関心事を横断的関心事として定義している。

本研究の手順として、ESS ロボットチャレンジ 2013 の競技ルールと Kobuki のハードウェアから仕様を考察する。考察した仕様をモデル化することで機能を明確にする。モデル化した仕様からアーキテクチャの構築に必要なコンポーネントを考察する。コンポーネント間の関係を考察する。考察した関係から自律走行ロボットのオブジェクト指向アーキテクチャを構築する。構築したアーキテクチャと仕様をもとに、Kobuki のオブジェクト指向アーキテクチャを構築する。関心事がアーキテクチャのどの部分を横断しているか考察する。これによってアスペク

ト指向アーキテクチャが構築できる。

2 背景技術

本章では本研究の背景技術を紹介する。

2.1 アスペクト指向技術

アスペクト指向技術とは、横断的関心事をアスペクトとしてモジュール化する技術である。横断的関心事とは、支配的構造によって得られる構造の複数のモジュールに散在する関心事である。横断的関心事をモジュール化することによってソフトウェアの再利用性、柔軟性が向上する。

2.2 E-AoSAS++

E-AoSAS++[2] は本研究室で提案している組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++ ではシステムを並行して動作する状態遷移機械(以下, CSTM)の集合と定義している。CSTM はイベントを受信することで状態遷移し、アクションとして CSTM の処理を実行する。その時に他の CSTM にイベントを送信する。これらの協調動作によって組込みシステムの機能を実現する。E-AoSAS++ では以下の 5 つを横断的関心事として定義している。

- 並行性
- 状態遷移
- エラー処理
- 耐故障性
- 実時間性

2.3 FORM

FORM[1] は仕様モデルを記述する手法として Kang らによって提案された。構成要素であるフィーチャを木構造で表現することにより、フィーチャが持つ役割を明確にする。FORM に基づくフィーチャモデルは組込みソフトウェア開発で代表的に用いられる仕様モデルである。FORM に基づくフィーチャモデルはフィーチャの役割を明確にするためにフィーチャを 4 つの層に分けて記述する。以下に 4 つの層とそれぞれの役割を示す。

- Capability Layer
 - ユーザの要求を実現する機能, 非機能
- Operating Environment Layer
 - ハードウェア, ソフトウェアの操作環境

- Domain Technology Layer
 - 特定のドメインで用いる技術
- Impremantation Technique Layer
 - ドメインを限らず用いることのできる技術

2.4 アーキテクチャスタイル

本研究では MVC とイベント駆動型アーキテクチャを組み合わせて用いる。

2.4.1 MVC

MVC は機能をデータの処理を行なう Model, 出力を行なう View, 入力によって Model, View を制御する Controller に分割するアーキテクチャスタイルである。機能を MVC で分割することによってコンポーネントごとの役割を明確にし、コンポーネントの独立性を向上することができる。

2.4.2 イベント駆動型アーキテクチャ

イベント駆動型アーキテクチャはあるコンポーネントがイベントを発行し、そのイベントに関心のあるコンポーネントがイベントを受け取ることで機能を実現するアーキテクチャスタイルである。コンポーネントをイベントごとに分割することで、コンポーネントごとの役割を明確にし、コンポーネントの独立性を向上することができる。

3 自律走行ロボットのオブジェクト指向アーキテクチャの構築

本章では自律走行ロボット Kobuki のオブジェクト指向アーキテクチャ構築について述べる。仕様の考察、必要となる機能の考察を行なうことで自律走行ロボットのオブジェクト指向アーキテクチャを構築し、Kobuki 本体の仕様を考慮することで Kobuki のオブジェクト指向アーキテクチャを構築する。

3.1 仕様の考察

本研究室は ESS ロボットチャレンジ 2013 に参加した。ESS ロボットチャレンジ 2013 ベーシック課題 [3] では、10 分以内に以下の条件を満たすことをルールとしている。

- 障害物、ドッキングステーションを含む平面の環境内の地図を作成する。
- 2 箇所のドッキングステーションを巡回し、ドッキングした状態で停止する。

Kobuki にはエンコーダ、タッチセンサー、IR センサー、段差センサー、タイヤ 1 つにつき 1 つ、計 2 つのモータが備えられている。これらの条件から必要となる機能を考察し、以下に示す。

- 速度制御
- 方向制御
- 地図作成
- 現在位置情報取得

- 周辺情報取得

3.2 オブジェクト指向アーキテクチャの構築

考察した機能をそれぞれの役割を持つコンポーネントとする。それらのデータの処理を行なうコンポーネントが存在すると考える。MVC における Controller から Model, Model から View へイベント送信が行われると考え、アーキテクチャを構築する。構築したアーキテクチャを図 1 に示す。構築したアーキテクチャにおけるコンポーネントと責務を表 1 に示す。

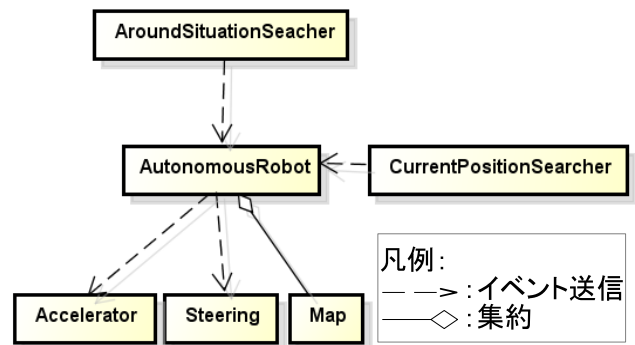


図 1 自律走行ロボットのオブジェクト指向アーキテクチャ

表 1 コンポーネントと責務

コンポーネント	責務
Accelerator	速度制御
AroundSituationSearcher	周辺情報取得
AutonomousRobot	データ処理
CurrentPositionSearcher	現在位置情報取得
Map	地図作成
Steering	方向制御

考察した機能から仕様を FORM に基づくフィーチャモデルを用いて記述した。記述した仕様モデルを図 2 に示す。

このアーキテクチャと仕様モデルを考慮することで、Kobuki のオブジェクト指向アーキテクチャを構築する。構築したアーキテクチャを図 3 に示す。

4 自律走行ロボットのアスペクト指向アーキテクチャの構築

3 章で構築した自律走行ロボット Kobuki のオブジェクト指向アーキテクチャから横断的関心事、関心事が横断するコンポーネントを考察し、アスペクト指向アーキテクチャを構築する。

4.1 横断的関心事の考察

Kobuki のオブジェクト指向アーキテクチャには、E-AoSAS++[2] で定義されている関心事の他にログを取得

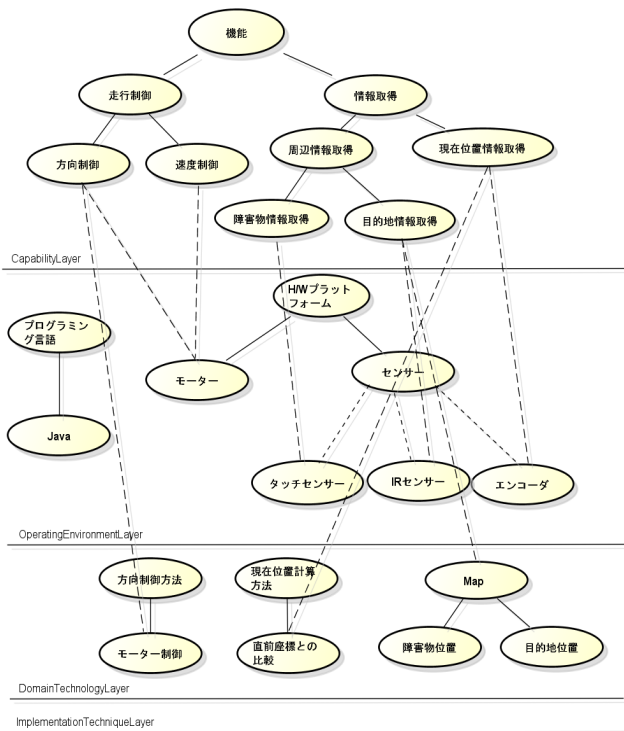


図2 Kobukiの仕様モデル

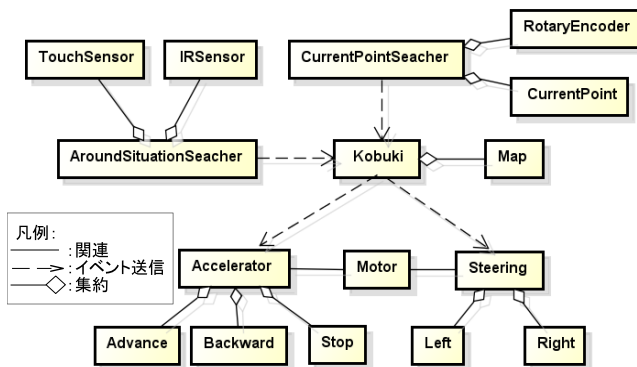


図3 Kobukiのオブジェクト指向アーキテクチャ

する関心事が横断していると考えられる。ログ取得は異常が発生している箇所を特定するために全てのコンポーネントで行なわれていると考えられることから、すべてのコンポーネントに横断していると考えられる。E-AoSAS++[2]では並行性、状態遷移を global concern としている。よって並行性、状態遷移は全体に横断していると言える。関心事と横断しているコンポーネントの関係を整理し、表2に示す。RealTime, FaultTolerance はそれぞれ実時間処理、耐故障処理を実現するコンポーネントを表す。

4.2 アスペクト指向アーキテクチャの構築

3章で構築したオブジェクト指向アーキテクチャ、整理した関心事とコンポーネントの関係をを用いてアスペクト指向アーキテクチャを構築する。構築したアーキテクチャを図4に示す。ステレオタイプで aspect と記述されている

表2 関心事とコンポーネントの関係

関心事	コンポーネント
並行性	全体
状態遷移	全体
エラー処理	RealTime FaultTolerance
耐故障性	AroundSituationSeacher CurrentPointSeacher
実時間性	Accelerator AroundSituationSeacher AutonomousRobot CurrentPointSeacher
ログ取得	全体

コンポーネントがアスペクトである。

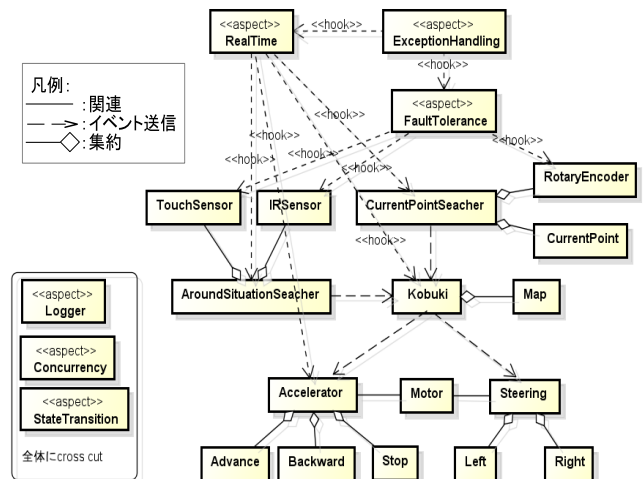


図4 Kobukiのアスペクト指向アーキテクチャ

5 考察

本章では本研究におけるアスペクト指向アーキテクチャの構築手順の妥当性と今後の課題について考察する。

5.1 アーキテクチャの構築手順の妥当性

本研究では以下の手順で自律走行ロボット Kobuki のアスペクト指向アーキテクチャを構築した。

- 要求である ESS ロボットチャレンジ 2013 の競技ルールと Kobuki のハードウェアから仕様を考察
- 仕様から一般的な自律走行ロボットのオブジェクト指向アーキテクチャを構築
- Kobuki の仕様モデルを記述
- 仕様モデルからコンポーネントと責務の関係を整理し、Kobuki のオブジェクト指向アーキテクチャを構築
- 横断的関心事と横断するコンポーネントを考察し、ア

スペクト指向アーキテクチャを構築

アーキテクチャは MVC, イベント駆動型アーキテクチャを組み合わせることでコンポーネントの独立性が向上し, 要求の変更に柔軟に対応できると考えられる. 仕様モデルは FORM[1] に基づくフィーチャモデルを用いた. FORM に基づくフィーチャモデルによる仕様の記述は組込みシステム開発において一般的に用いられているので本研究の事例における仕様の記述法として妥当であると言える. 関心事は E-AoSAS++[2] で定義されているものとログの取得を考慮した. E-AoSAS++ は組込みシステムのためのソフトウェアアーキテクチャスタイルで, 自律走行ロボットの関心事の考察に適していると考えられる. ログの取得は解析性を向上させ, 開発過程や故障時のコスト削減に有用であると考えられる.

5.2 今後の課題

本研究では ESS ロボットチャレンジ 2013 の課題を達成するためのアスペクト指向アーキテクチャの構築を行った. 本研究におけるアスペクト指向アーキテクチャの構築手順の一般性を検証するために同様の構築手順を用いて他の自律走行ロボットについて検証する必要がある. 今回作成した FORM に基づくフィーチャモデルでは, Imprementation Technique Layer にフィーチャが現れなかった. Imprementation Technique Layer にフィーチャが現れる場合の検証, もしくは仕様の記述方法を検討する必要がある. これらの課題をすることにより, 自律走行ロボットのアーキテクチャの変動点について考察が可能となり, 自律走行ロボットのプロダクトラインを生成することが可能になると考える. プロダクトラインを生成することで, 岡田による研究 [4] を用いて自律走行ロボットのプロダクトアーキテクチャの自動構築が可能になると考える. また, 今回 ESS ロボットチャレンジ 2013 の課題を達成するための Kobuki のアスペクト指向アーキテクチャを構築する過程で, 自律走行ロボットのオブジェクト指向アーキテクチャを構築した. このアーキテクチャを Kobuki の事例に基づき, 詳細化することで Kobuki のオブジェクト指向アーキテクチャを構築した. Kobuki のアスペクト指向アーキテクチャを抽象化することで, 一般的な自律走行ロボットのアスペクト指向アーキテクチャを構築した. 構築した一般的な自律走行ロボットのアスペクト指向アーキテクチャを図 5 に示す. このアスペクト指向アーキテクチャは本研究における事例でのみ妥当性が確認されている. 今後の課題として, 他の自律走行ロボットの事例を用いた妥当性の確認を行わなければならないことが挙げられる. 今後このアーキテクチャについて Kobuki 以外の事例を用いて妥当性の確認ができれば, 一般的な自律走行ロボットのアスペクト指向アーキテクチャとして適切であると言える.

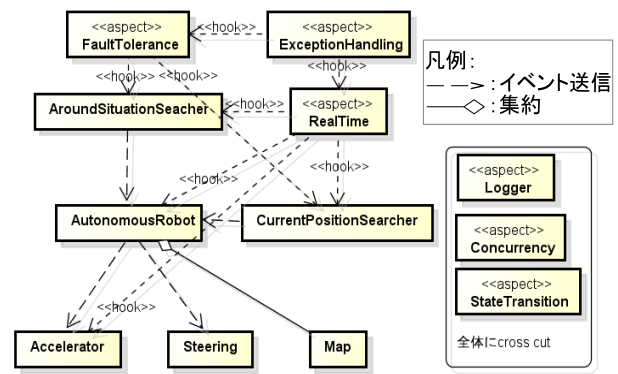


図 5 自律走行ロボットのアスペクト指向アーキテクチャ

6 おわりに

本研究では ESS ロボットチャレンジ 2013 の課題とハードウェアから仕様を考察し, 仕様をモデル化した. MVC とイベント駆動型アーキテクチャを組み合わせることで, 仕様から一般的な自律走行ロボットのオブジェクト指向アーキテクチャの構築を行った. 一般的な自律走行ロボットのアーキテクチャと仕様モデルから, Kobuki のオブジェクト指向アーキテクチャの構築を行った. 構築したオブジェクト指向アーキテクチャから, 関心事が横断するコンポーネントを考察した. オブジェクト指向アーキテクチャと関心事から, Kobuki のアスペクト指向アーキテクチャを構築した. 今後の課題として以下の内容が挙げられる.

- Imprementation Technique Layer にフィーチャが現れる場合の検証
- 仕様の記述方法の検討
- 他の自律走行ロボットでの事例検証

参考文献

- [1] K. C. Kang, S. Kim, J. Lee, K. Kim, and E. Shin, "FORM: A Feature-Oriented Reuse Method with Domain-Specific Reference Architectures," *Annals of Software Engineering*, vol. 5, no. 1, pp. 143-168, 1998.
- [2] M. Noro, A. Sawada, Y. Hachisu, and M. Banno, "E-AoSAS++ and its Software Development Environment," *Proceedings of the 14th Asia-Pasific Software Engineering Confarence(APSEC2007)*, pp. 206-213, 2007.
- [3] ESS ロボットチャレンジ実行委員会, "ESS ロボットチャレンジ 2013 競技概要," <http://www.qito.kyushu-u.ac.jp/ess/2013/rule2013.Ver.1.pdf>, 2013.
- [4] 岡田大輝, "SOA に基づくシステムのためのアプリケーションプラットフォームのプロダクトラインに関する研究," 南山大学大学院数理情報研究科, 2014.