自動制御と操作性を考慮した組込みソフトウェアのアーキテクチャの提案

―飛行船制御ソフトウェアを例題として―

2009SE286 坪井智哉

指導教員：野呂昌満

1　はじめに

制御対象機器の複雑化ならびにユーザ要求の多様化に伴い，組込みソフトウェアにおいても大規模化，複雑化が進んでいる．その開発にはさらなる生産性の向上と信頼性の保証が求められている．組込みソフトウェアはシステムを状態遷移機械の集まりと捉えるので，オブジェクト指向技術を支配的分割として用いた構造の定義が重要である．組込みソフトウェアには支配的分割に対して横断的関心事が散在している．横断的関心事をモジュール化する技術としてアスペクト指向技術がある．横断的関心事をアスペクトとしてモジュール化することで，再利用性や保守性を高めることができる．本研究室では組込みソフトウェア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++)[1] を提案している．E-AoSAS++では並行性，実時間性，耐故障性などの非機能特性を横断的関心事として特定している．

組込みソフトウェアには，我々が特定した非機能特性以外にも横断的関心事は存在する．自動制御や操作性は支配的分割に横断するので，分離する必要がある．

本研究の目的は，アスペクト指向アーキテクチャの構築支援である．自動制御や操作性を横断的関心事として捉え，アスペクトとして分離するためのアスペクト指向アーキテクチャを定義する．

本研究では MVC やプロセス制御などの既存のアーキテクチャスタイルを用いて自動制御や操作性を表現できると仮定した．飛行船制御ソフトウェアのアーキテクチャに複数のアーキテクチャスタイルを適用した．アーキテクチャスタイルの構成要素が横断する部分を特定し，アスペクトとして分離した．適用したアーキテクチャスタイルと E-AoSAS++ を融合し，アスペクト指向アーキテクチャを提案した．融合とは，一方のスタイルを他方のスタイルの記述方法で記述することを指す．

本研究の成果として，自動制御や操作性をアスペクトとして分離したアーキテクチャを定義した．結果，アスペクト指向アーキテクチャの構築支援が可能になった．

2　背景技術

2.1　E-AoSAS++

E-AoSAS++[1] 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ソフトウェアアーキテクチャスタイルである．E-AoSAS++ では，システムを並行に動作する状態遷移機械（以下，CSTM）の集合として定義している．CSTM は受信したイベントに応じて状態を遷移し，状態遷移時にアクションとして CSTM の処理を実行する．その際に他の CSTM にイベントを送信する．この各 CSTM 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++ に基づくアーキテクチャ記述 [3] は UML を用いる．E-AoSAS++ におけるアスペクトの表現はステレオタイプを用いて UML の意味を拡張して表現する．

2.2　MVC スタイル

アーキテクチャに MVC を適用することで，構造をモデル・ビュー・コントローラの 3 つの要素で分割し，再利用性・保守性の高いソフトウェアを設計することができる．モデル・ビュー・コントローラの責務は次のように定義されている．モデルはアプリケーションの中核機能とデータに関する処理，ビューはユーザへの情報の表示，コントローラはユーザ入力をイベントとして受け取りモデル，ビューへのリクエストに変換する責務を持つ．本研究では，操作性に関する構造を MVC を用いて表現可能と仮定する．

2.3　プロセス制御スタイル

プロセス制御スタイルは，連続的な入力の変化を動的に監視し，目標に応じた出力を維持するためのものである．構成要素は制御部分と処理部分である．本研究では，自動制御に関する構造をプロセス制御を用いて表現可能と仮定する．

3　飛行船のアーキテクチャ構築

ESS ロボットチャレンジ 2012 の仕様 [2] に基づいた飛行船制御システムを事例とする．飛行船制御ソフトウェアに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が求められる．横断的関心事を特定するために，飛行船制御ソフトウェアをオブジェクト指向技術で支配的分割し，オブジェクト指向アーキテクチャを設計した．MVC とプロセス制御を飛行船のアーキテクチャに適用した．

3.1　アーキテクチャスタイルの適用

飛行船のオブジェクト指向アーキテクチャに MVC，プロセス制御を適用，分割した．アーキテクチャスタイルの各要素が持つ責務と各コンポーネントの責務を比較し，分割した．図 1 は 2 つのアーキテクチャスタイルで分割した飛行船のアーキテクチャである．非機能特性として実時間

性を考慮した．

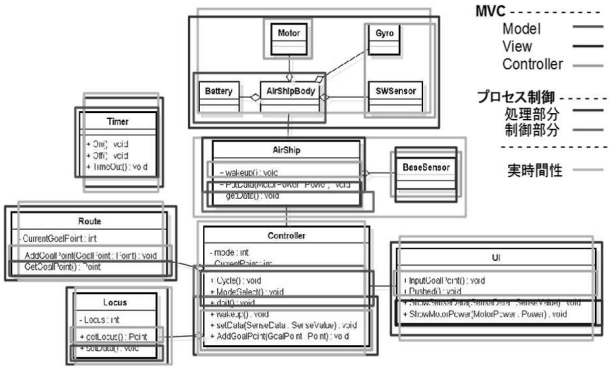

図 1　分割した飛行船のアーキテクチャ

3.2　横断的関心事の分離

アーキテクチャスタイルの構成要素が持つ責務と各コンポーネントの責務を比較し，アスペクトを抽出した．MVC の Controller，プロセス制御の制御部分，実時間性は複数のコンポーネントに横断しているので，アスペクトとして抽出した．図 2 は横断的関心事を分離した飛行船のアーキテクチャである．

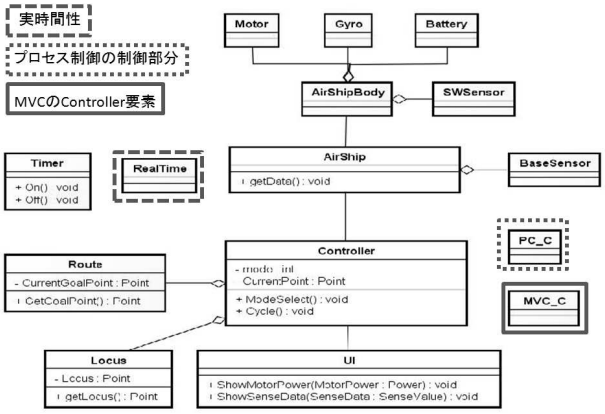

図 2　横断的関心事を分離した飛行船のアーキテクチャ

4　アスペクト指向アーキテクチャの提案

抽出したアスペクトを E-AoSAS++ の記述方法 [3] で表現し，実時間性，操作性，自動制御を考慮したアスペクト指向アーキテクチャを提案した．図 3 はアスペクトとして実時間性，MVC の Controller，プロセス制御の制御部分があることを示す．E-AoSAS++ の記述方法で記述したコンポーネント図の基本的な構成要素の意味を以下に示す．

- View：システムの全体，または一部を表現
- Aspect：View によって分割されたモジュール
- IAD(アスペクト間記述)：横断的関心事の振舞い
- AspectCoordinator : Aspect の構成を管理するメタ状態遷移機械

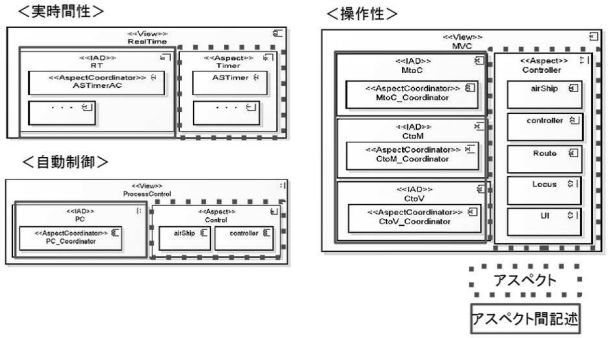

図 3　提案するアスペクト指向アーキテクチャ

5　考察

提案したアーキテクチャの有用性と本研究のアプローチの妥当性について考察する．複数のアーキテクチャスタイルを適用し，自動制御と操作性に関する横断的関心事をアスペクトとして分離した．分離したアスペクトに対応したアスペクト指向アーキテクチャを定義した．定義したアーキテクチャを用いることで，系統的にアスペクトを抽出したアーキテクチャを構築することが可能になった．他の横断的関心事についてもアーキテクチャスタイルで構造を表現可能であるとき，本研究のアプローチを用いることでアスペクト指向アーキテクチャを定義できると考えた．結果，本研究のアプローチは妥当であると考える．

6　おわりに

本研究では，自動制御と操作性を横断的関心事として分離するために，MVC スタイルとプロセス制御スタイルを用いてアスペクト指向アーキテクチャを定義した．適用したスタイルと E-AoSAS++ を融合した結果，アスペクト指向アーキテクチャを定義することができた．今後の課題として，本研究で定義したアーキテクチャを他の組込みシステムに適用し，同様に自動制御と操作性をアスペクトとして分離可能か検証する必要がある．

参考文献

[1] M. Noro, A. Sawada, Y. Hachisu, and M. Banno, “E-AoSAS++ and its Software Development Environment,” *Proceedings of the 14th Asia-Pacific Software Engineering Conference(APSEC2007),* pp. 206-213, 2007.

[2] ESS ロボットチャレンジ 2012 実行委員会，”ESS ロボットチャレンジ競技規定，” http://www.sigemb.jp/rc/2012/ess_rc2012rule.pdf, 2012.

[3] 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，” 知能ソフトウェア工学研究会 (KBSE)，vol. 108，no. 449，pp. 55-60，2009.