

組み込みソフトウェアのための アスペクト指向アーキテクチャスタイルの提案

M2005MM002 坂野 将秀

指導教員 野呂 昌満

1 はじめに

ソフトウェア工学の歴史はモジュール化の歴史といっても過言ではない。機能をモジュール化する手続き指向、データ構造と機能を1つのモジュールとするデータ抽象化、近年では、オブジェクト指向が提案されている。オブジェクトモジュールは、ほとんどの要求を適切にモジュール化できると言われている。しかし、オブジェクト指向ではコンサーン横断問題を解決できない。

組み込みシステムはマイクロプロセッサの進歩にともない、ハードウェアインテンスシステムであったものから、ソフトウェアインテンスシステムになってきている。これにともない、組み込みソフトウェアの規模は増大し、制御ソフトウェアの構造は複雑化してきている。近年では、機能のライブラリ化をおこない、オペレーティングシステムのシステムコールを整理することで、開発支援することが試みられているが、これにも限界がある。

このような背景のもと、組み込みソフトウェアの構造を適切にモジュール化のためのアーキテクチャを構築する必要があるとの認識を、我々は持つ。組み込みソフトウェアの特性を考えた場合、アスペクト指向技術の導入は不可避だと考え、そのための主要課題は以下の2つである。

1. コンサーンの抽出、整理
2. コンサーンの取り扱い方法の定義

前者は、組み込みソフトウェアの構造整理において、機能特性と同時に実時間性、並行性、耐故障性などの非機能特性考慮する必要を指す。後者は、コンサーンの取り扱い方法が確立されていないことに起因する。抽出したコンサーンを適切にモジュール化する方法を整理する必要がある。

本研究は、コンサーン横断問題を考慮した、体系的なアーキテクチャの記述方法を提案することを目的とする。組み込みソフトウェアにおけるアスペクト指向アーキテクチャスタイル(以下、E-AOSAS++)を提案する。E-AOSAS++は、組み込みソフトウェアのアーキテクチャを、統一的かつ適切にモジュール化する記述方法を実現することを目標とする。

本研究における問題解決のアプローチとして、実用性、理論的考察の両方を重視した。実用性を保証するために、事例から複数のアーキテクチャを構築し、抽象化することでスタイルを構築するボトムアップ方式の立場をとった。さらに理論的考察のために、構築したアーキテクチャスタイルのあるべき姿を検証するトップダウンの方式の立場をとった。

2 XCC モデル

我々は、E-AOSAS++の理論的整合性結果の必然性のために、X-Aspect Oriented Architecture Style Construction Model based on Component-Connector Model(以下、XCCモデル)を定義した。XCCモデルとはアスペクトアーキテクチャスタイルを定義するモデルである。XCCモデルでは、アーキテクチャスタイルにアスペクトを取り入れる場合の、コンポーネントとアスペクトの関連、コネクタとアスペクトの関連を整理することで、アーキテクチャスタイルを構築する。

我々は、XCCモデルによってアーキテクチャスタイルの構築手順を標準化できると考える。

2.1 XCCモデル概要

XCCモデルは3段階のステップを定義する。図1はXCCモデルの概要を示したものである。

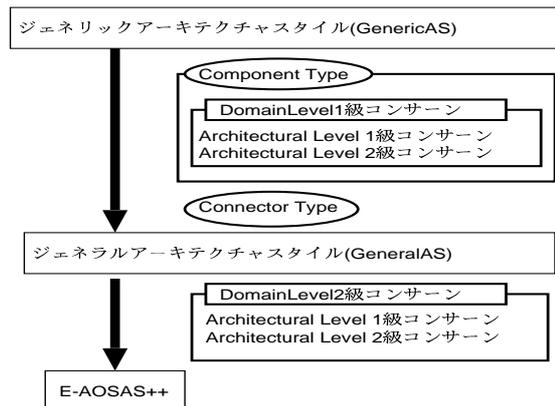


図1 XCCモデル概要

XCCモデルでは、GenericAS、GeneralAS、E-AOSAS++の順に構築していく。本研究では、Shaw、Garlanが提案しているコンポーネント、コネクタモデルを原始的なアーキテクチャスタイルと考え、GenericASとして定義する。GenericASからGeneralAS間には、コンポーネントタイプ、コネクタタイプ、GeneralASからE-AOSAS++はコンサーンをパラメータとして与えることで構築する。

XCCモデルの3段階のステップを定義する理由は、アーキテクチャスタイルの具象化する手順を明確に定義するためである。GenericASから、コンポーネントの階層化方法を定義したGeneralASを定義する。GeneralASから、各コンポーネントの特性を定義したE-AOSAS++を定義する。コンポーネントの階層化を実現し、各コンポーネントの特性を決定することで、アーキテクチャスタイル構築手順を明確に定義できると考えた。

2.2 コンサーン

本研究でとりあげるコンサーンについて説明する．コンサーンを分類するさい，2つの尺度からコンサーンを分類することができる考えた．定義した尺度は以下の2つである．

- Architecture Level コンサーン
- Domain Level コンサーン

Architecture Level コンサーン ,Domain Level コンサーンは，我々が過去いくつかのアーキテクチャを構築してきた経験から定義した尺度である．我々は，Architecture Level コンサーンをソフトウェアの構造に関する尺度，Domain Level コンサーンをソフトウェアの振舞いに関する尺度と考えた．

Architecture Level コンサーン ,Domain Level コンサーンの2つの尺度には直交した関係であることを確認した．図2は2つのコンサーン分類尺度の関連を示したものである．

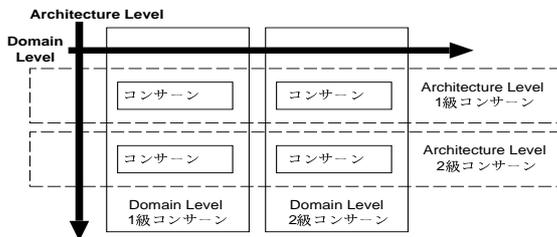


図2 コンサーン分類尺度の関連

2.3 ジェネリックアーキテクチャスタイル

GenericAS について説明する．GenericAS モデルは，XCC モデルにおける基本となるアーキテクチャスタイルである．各コンポーネントは，コネクタを介して他のコンポーネントにメッセージを通知する．GenericAS に以下のコンポーネントタイプ，コネクタタイプを加えることで，GeneralAS を定義する．

- コンポーネントタイプ
 - Primitive Component
 - Configuration Composite Component
 - Aggregation Composite Component
- コネクタタイプ
 - Connector
 - Multicast Connector
 - Unicast Connector

2.4 ジェネラルアーキテクチャスタイル

定義した GeneralAS について説明する．GenericAS にコンポーネントタイプ，コネクタタイプを与えることで，具象化したアーキテクチャスタイルが構築できると考えた．GeneralAS は，GenericAS と比べコンポーネントの階層化，階層化にともなうメッセージ通知の方法を考慮しアーキテクチャスタイルである．構築した GeneralAS を図3に示す．

GeneralAS では，ソフトウェアアーキテクチャを PComponent ,CCComponent ,ACComponent の協調動作

で記述する．CCComponent は，システムの構成を管理(以下，CC)する複合コンポーネントである．ACComponent は，複数のコンポーネントが協調動作する複合コンポーネントを実現する．

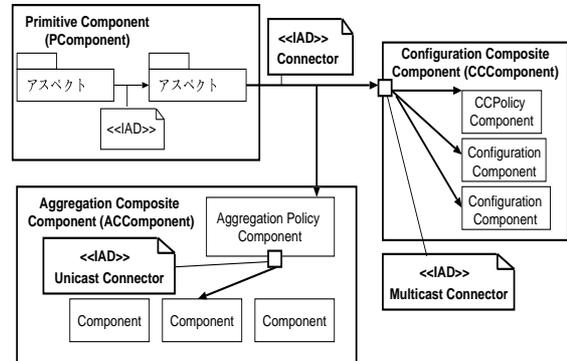


図3 ジェネラルアーキテクチャスタイル (GeneralAS)

GeneralAS に Domain Level2 級における Architecture Level1, 2 級コンサーンをパラメータとしてあたえることで E-AOSAS++ を提案する．各コンサーンは我々が過去組み込みソフトウェアのアーキテクチャを取り扱ってきた経験から抽出したコンサーンである．パラメータとしてあたえたコンサーンを以下に示す．

- Architecture Level1 級コンサーン
 - CC コンサーン:通常 CC，例外処理，耐故障性，実時間処理
- Architecture Level2 級コンサーン
 - 並行処理，状態遷移，コア

3 組み込みソフトウェアにおけるアスペクト指向アーキテクチャスタイル (E-AOSAS++)

E-AOSAS++ は，組み込みソフトウェアのアーキテクチャを構築するうえの，系統的な記述方法である．E-AOSAS++ は，前身である E-AOSAS+ の基本概念に，前章で示した，コンポーネント，コネクタの概念を加えたものである．E-AOSAS+ は，組み込みソフトウェアは並行に動作する状態遷移機械の集合とすることを基本概念としている．E-AOSAS++ では，以下の3つが主要概念となる．

- 並行状態遷移機械 (CSTM)
- Configuration CompositeCSTM (C-CCSTM)
- Aggregation CompositeCSTM (A-CCSTM)

3.1 CSTM

E-AOSAS++ における原始コンポーネントは，並行状態遷移機械 (以下，CSTM) である．CSTM は，イベント通知を受理することで動作し組み込みソフトウェアにおける機能を実現している．CSTM の構造を図4に示す．CSTM では，並行処理アスペクト，状態遷移アスペクト，コアアスペクトの3つのアスペクトとアスペクト間の関連を記述する，IAD で構成される．各アスペクトは，互いに IAD を介して，連動動作をする．アスペクト間の関連は，IAD に記述されているので，各アスペクト

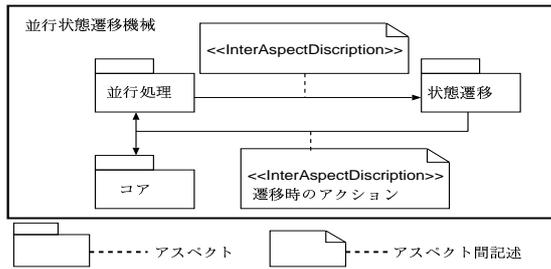


図 4 CSTM

を変更した場合でも、他のアスペクトに変更がおよぶことはない。並行処理アスペクトは、状態遷移機械の、起動、停止などの同期処理をおこなう、コンポーネントの集合である。状態遷移アスペクトは、CSTM の状態遷移に関するコンポーネントの集合である。コアアスペクトは、CSTM における核となる機能を実現するコンポーネントの集合である。IAD アクションは、CSTM 間のコネクタとしても定義される。CSTM の典型的動的挙動を図 5 に示す。

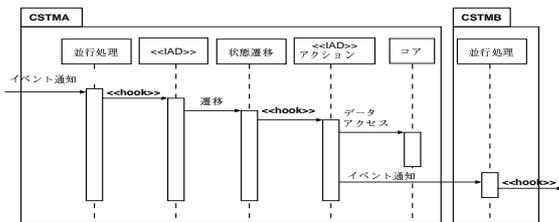


図 5 CSTM における典型的動的挙動

3.2 Configuration Composite CSTM

C-CCSTM はコンフィギュレーションコントロール (以下、CC) コンサーンを実現した複合並行状態遷移機械である。GeneralAS における、コンフィギュレーション複合コンポーネントと同様に、システムの構成管理を実現するコンポーネントである。CC コンサーンは CC ポリシーコンサーン、コンフィギュレーションコンサーンで構成される。CC ポリシーコンサーンを、コンフィギュレーションを切り替えるさいの、ポリシーに関連するコンサーンである。コンフィギュレーションコンサーンは、システムの構成を規定しているコンサーンである。

C-CCSTM は、CC ポリシー CSTM と、コンフィギュレーション CSTM で構成される。CC ポリシーコンサーン、コンフィギュレーションコンサーンをそれぞれ実現したものが、CC ポリシー CSTM、コンフィギュレーション CSTM である。CC ポリシー CSTM は、コンフィギュレーション CSTM の active(sleep) 状態を管理することで、システムのコンフィギュレーションを切り替える。active(sleep) 状態を切り替えるさい、CC ポリシー CSTM は、active 通知、sleep 通知をおこなうことで実現する。

E-AOSAS++ において例外処理コンサーン、通常機能 CC コンサーン、実時間処理コンサーン、耐故障性コ

ンサーン、C-CCSTM によって記述できると我々は考えた。

3.3 Aggregation Composite CSTM

A-CCSTM は、複数の CSTM が連動して動作することで 1 つの機能を実現する、複合並行状態遷移機械である。GeneralAS における、アグリゲーション複合コンポーネントにおける、複数のコンポーネントが連動して動作することで機能を実現するコンポーネントである。複数のコンポーネントが連動して動作することで 1 つの機能を実現することは、組込みソフトウェアにとって必要な概念と考えた。

A-CCSTM は、アグリゲーションポリシー CSTM、各原始機能を実現する CSTM で構成される。アグリゲーションポリシー CSTM は、各 CSTM の連動動作を実現する。このさいアグリゲーションポリシーコンサーンは、ユニキャストコネクタをもちいることで、各 CSTM に適切にイベント通知をおこなう。

4 考察

構築した E-AOSAS++ を本研究の目的である以下の 2 点から考察する。

- 適切なモジュール化についての考察
- 統一的な記述方法についての考察

4.1 適切なモジュール化

E-AOSAS++ をもちいることで、組込みソフトウェアのアーキテクチャ構築時に適切なモジュール化が可能であるかを考察する。事例として組込みシステムのネットワーク制御におけるログシステムを、オブジェクト指向設計した場合と、E-AOSAS++ をもちいて設計した場合を比較する。モジュール化の比較を図 6 に示す。

E-AOSAS++ をもちいた場合、オブジェクト指向設計した場合の変更箇所の違いの原因としては、OutPutDevice の役割りのモジュール化方法の違いが原因と我々は、考察する。OutPutDevice の役割りは以下のとおりである。

- Port にたいして、メッセージを通知する。
- Buffer にたいして、メッセージを通知する。
- メッセージの通知先を切り替える。

オブジェクト指向設計で OutPutDevice を考慮したさい、抽出した 3 つの役割は、1 つのコンポーネントとして実現したのに対して、E-AOSAS++ では、3 つのコンポーネントに分割されている。E-AOSAS++ では、OutPutDevice の役割りごとに InActiveOutPutDevice、ActiveOutPutDevice、ログシステム CCPolicyCSTM の 3 つの CSTM で実現している。オブジェクト指向設計した OutputDevice は、複数の役割りを 1 つのコンポーネントとしたために、複数のコンポーネントとの関連をもつことになる。このようなコンポーネントに変更が及んださい、変更が及ぶ影響箇所が増大する。E-AOSAS++ で記述した OutPutDevice を複数のコンポーネントに分割することで、ソフトウェア

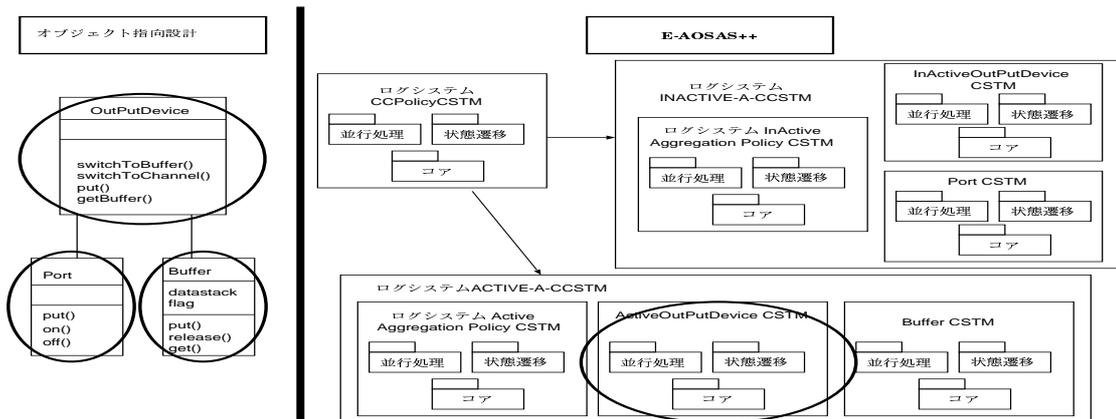


図6 ログシステムにおけるモジュール化の考察

の柔軟性を確保している。我々は、その他の事例として、走行支援ソフトウェア、エンジン制御ソフトウェアにも E-AOSAS++ を適用しアーキテクチャを適切にモジュール化できることを確認した。

4.2 統一的な記述方法

E-AOSAS++ におけるコンポーネント、コネクタの記述方法について、考察する。E-AOSAS++ で提案されている記述方法を、図7に示す。

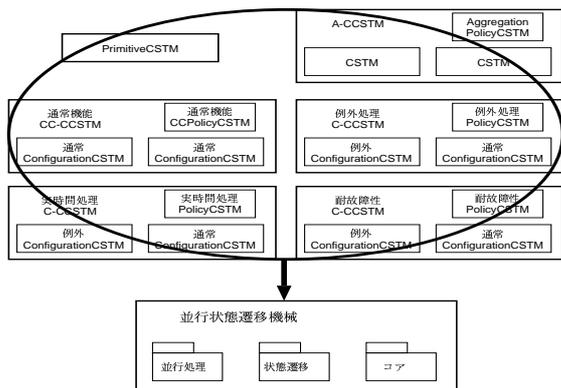


図7 CSTM, 複合 CSTM の記述方法

E-AOSAS++ において定義されている記述方法は、すべて CSTM もしくは、複合 CSTM でのみ定義されている。CSTM は、並行処理アスペクト、状態遷移アスペクト、コアアスペクト、各アスペクトの関連を示した IAD で記述される。通常機能 CC、例外処理、耐故障性、実時間処理の取扱い方法は、複合 CSTM として記述される。

E-AOSAS++ における記述方法は、組み込みソフトウェアは、並行状態遷移機械の協調動作で定義できるという最小の基本概念で整理されている。少ない基本概念で記述された組み込みソフトウェアのアーキテクチャは、構造のパターン、動的挙動のパターンなどが定義できる。E-AOSAS++ における、構造のパターンを図8に示す。

各 CC コンサーンは、CCPolicyCSTM, ConfigurationCSTM で構成される C-CCSTM で実現される。E-AOSAS++ を適用するドメインにおいて、CC コンサ

ーンが存在する場合、この共通構造はパターンとしてあらわれると考えられる。

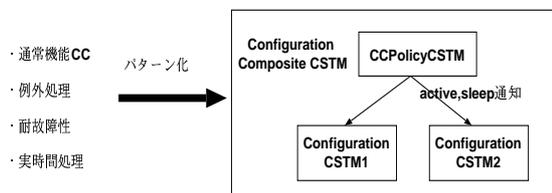


図8 E-AOSAS++ における記述方法のパターン (構造)

E-AOSAS++ における動的挙動のパターンも存在する。CSTM は、常に、並行処理アスペクト、状態遷移アスペクト、コアアスペクトの順に連動動作する、動的挙動のパターンが存在する。これらは、UML 図におけるステートチャート図から、大部分記述することが可能である。

5 まとめ

本研究では、E-AOSAS++、XCC モデルの提案と妥当性について考察をおこなった。E-AOSAS++、XCC モデルを提案することで、組み込みソフトウェアの系統的開発をおこなうさいの核となる、アーキテクチャスタイルを提案した。

参考文献

- [1] C. Angelov, N. Marian, K. Sierszecki, and J. Ma, "Model-Based Design and Verification of Embedded Software," *Proc. of the 5th European Workshop on Research and Education in Mechatronics REM'2004*, 2004.
- [2] J. A. D. Pace, and M. R. Campo, "ArchMatE: From Architectural Styles to Object-Oriented Models through Exploratory Tool Support," *Proc. OOPSLA '05*, 2005.
- [3] P. A. Hsiung, and C. Y. Lin, "VERTAF: An Application Framework for the Design and Verification of Embedded Real-Time Software," *Proc. IEEE Transaction on Software Engineering*, 2004.