

# 組込みソフトウェア開発環境に関する研究 —プラットフォームコードの再設計—

2009SE300 渡邊晃

指導教員：沢田篤史

## 1 はじめに

近年、要求の多様化などに伴い、システムの大規模化や複雑化が急速に進んでおり、生産性の向上が求められている。組込みソフトウェア開発においては、非機能要求に関する関心事が絡み合うので、適切なモジュール分割を行うことが困難となっている。ソフトウェアに散在する関心事をアスペクトとして抽出してモジュール化するアスペクト指向技術がある。アスペクト指向技術によりモジュール化を行うことで柔軟性、再利用性に優れたソフトウェアを開発することができる。

本研究室では組込みソフトウェア開発のためのアスペクト指向アーキテクチャスタイル（以下、E-AoSAS++）[2]を提案している。E-AoSAS++ではシステムを並行に動作する状態遷移機械の集合と定義し、並行性や状態遷移、リアルタイム性などの関心事を横断的関心事として特定している。E-AoSAS++には組込みソフトウェア開発における定形コードであるプラットフォームコードがあり、開発を支援している。

近年の組込みシステム開発の傾向としてクラウドシステムへの適応がある。クラウドシステムを適用することによって組込みシステムの保持する情報をクラウドシステムへ送信して、クラウド上で処理、分析を行い組込みシステムの制御に活用することができる。かつてはハードリアルタイムシステムである組込みシステムが処理に時間を要する外部との通信を制御に用いることは困難であった。ネットワークの通信速度やCPUの処理速度の向上によってリアルタイム性の要求を満たし、外部との通信によって制御を行うことが可能となっている。

組込みとは異なるドメインとして、ソフトウェアをサービスの集合として捉え、ソフトウェアをネットワーク上で連携させることによってシステムを構築するサービス指向アーキテクチャ（以下、SOA）がある。またサービス間の通信をルーティングによって仲介を行うことでサービスを連携させるSOAに基づくアプリケーションプラットフォーム[1]がある。

現在のプラットフォームコードだけではクラウドシステムに適応することが困難である。SOAに基づくアプリケーションプラットフォームに着目し、組込みシステムに適したアプリケーションプラットフォームを用いる必要がある。

本研究の目的は、組込みシステムとクラウドシステム間の通信の仲介に用いるアプリケーションプラットフォームを構築することである。

E-AoSAS++で特定されている横断的関心事の一つであるリアルタイム性に着目し、実行効率に優れたアプリケーションプラットフォームの構築を行う。

## 2 背景技術

### 2.1 E-AoSAS++

E-AoSAS++は本研究室で提案している組込みソフトウェア開発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。プラットフォームコードなどのソフトウェア開発を支援する環境を提案している。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はイベントを受け取る状態を遷移する。状態遷移時にアクションを実行し、その際に他のCSTMにイベントを送信して状態を遷移させる。このCSTMの協調動作によって組込みシステムを実現する。E-AoSAS++のプラットフォームコードは並行処理アスペクト、状態遷移アスペクト、インスタンス処理アスペクト、アクションアスペクトと各アスペクトを繋げるアスペクト間記述（IAD）によって構成されている。

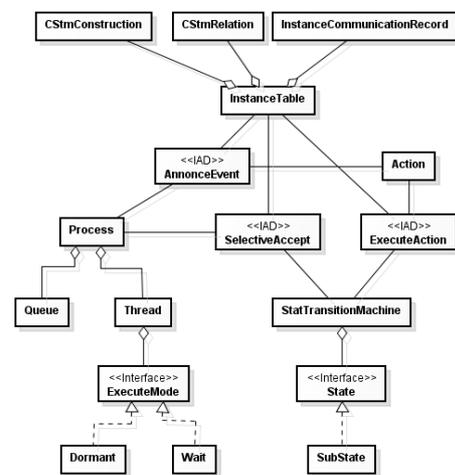


図1 E-AoSAS++のプラットフォームコード

### 2.2 SOA

SOAとは、システムをビジネスの業務処理ごとに分割した単位をサービスとして、ネットワーク上でサービスを連携させることによって大規模システムを構築する設計手法である。

### 2.3 SOAに基づくアプリケーションプラットフォーム

本研究室ではSOAのサービス間のメッセージ通信を仲介する機能を提供するアプリケーションプラットフォーム

(以下, App.PF) を提案している. SOA に基づく App.PF には PointToPoint 通信, PubSub 通信によるメッセージングやルーティングによる負荷分散などの機能がある.

### 3 実行効率に関する要求に基づくアプリケーションプラットフォームの構築

#### 3.1 アプリケーションプラットフォームの構築

App.PF の実装を行う際, 以下の図 2 のアーキテクチャに基づいて実装を行う. 実行効率を重視して実装することから, 既存の App.PF のアーキテクチャから Router を除いたアーキテクチャの構築を行った.

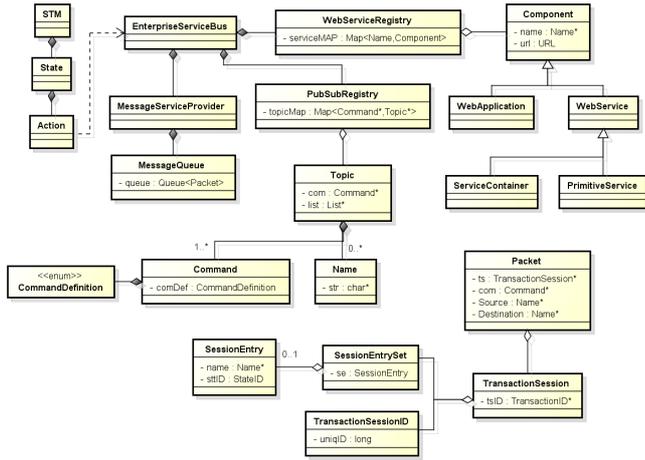


図 2 App.PF のアーキテクチャ

#### 3.2 試作した App.PF の概要

上記の図 2 に基づいて App.PF の試作を行った. 試作した App.PF の概要を以下の表 1 にまとめる. ミドルウェアの利用の有無によってミドルウェアへの対応の有無が決定される. ルーティングによる負荷分散の利用の有無によってアーキテクチャ上の Router の有無が決定される.

表 1 試作した App.PF の概要

要素名	選択肢, 手法
言語	C 言語を用いる
ミドルウェア	利用を考慮しない
ルーティング	負荷分散を考慮しない
プログラム	処理ステップ数の削減
プログラム	メモリアクセス量の削減

### 4 オブジェクト指向プログラミングに基づく App.PF の実装

アプリケーションプラットフォームの実装はオブジェクト指向プログラミングに基づき, C 言語を用いて行う. C 言語においてオブジェクト指向プログラミングの各要素を表現する際, 以下の表 2 に記した方法に基づいて行う.

表 2 C 言語におけるオブジェクト指向の表現方法

要素名	表現方法
クラス	汎用ポインタと関数ポインタを持つ構造体
カプセル化	汎用ポインタを属性へのアクセスに用いる
インスタンス	構造体型のポインタ変数
コンストラクタ	インスタンス生成用関数の引数を属性に代入
メソッド	関数
メッセージ	関数ポインタ

### 5 考察

構築した App.PF 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. 実行効率に優れた App.PF を構築するために, 実装する際に実行効率に関わる要素を整理した. C 言語においてオブジェクト指向プログラミングを表現する方法を整理し, App.PF の実装を系統的に行った. 構築した App.PF を用いることで組込みシステムに App.PF を適用することが可能になったと考える. 結果, 本研究のアプローチは妥当であると考えた.

本研究で構築した App.PF の問題点として, ミドルウェアの利用やルーティングによる負荷分散を考慮しなかったことから機能面において不十分である. 実行効率と機能面の要求を同時に満たす App.PF を構築する必要があると考える.

### 6 おわりに

本研究では, クラウドシステムに適応可能なプラットフォームコードを提案するために, E-AoSAS++ で特定されている関心事であるリアルタイム性に着目し, 実行効率に優れたアプリケーションプラットフォームの構築を行った. 今後の課題として, 本研究で構築したアプリケーションプラットフォームを E-AoSAS++ のプラットフォームコードに適用し, クラウドシステムに適応可能であるか検証する必要がある.

### 参考文献

- [1] 江坂篤侍, 野呂昌満, 沢田篤史, "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化に関する研究," 情報処理学会研究報告, 2013.
- [2] 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.