

コンテキストウェアネスを考慮した 組み込みソフトウェアアーキテクチャの設計と評価 — 運転支援アプリケーションを題材として —

2013SE153 大橋諒 20013SE218 寺本智哉

指導教員：沢田篤史

1 はじめに

近年，組み込みシステムではセンサの高性能・高機能化が進み，外部環境などのコンテキスト情報を正確に状態監視，判断が実現できるようになった．それに伴い，コンテキストウェアな処理を実現する組み込みシステムが増加してきた．一般に組み込みシステムはオブジェクト指向設計に基づいて設計され，コンテキストはハードウェア全体に横断する関心事である．組み込みシステムの開発ではリアルタイム性などの非機能特性が重要視される．リアルタイム性は指定されたデッドラインまでに処理を実行できる性質である [4]．このリアルタイム性はコンテキストと特定の処理の一部に横断する関心事となる．

コンテキストとリアルタイム性を考慮した場合，コンテキストに応じたアクチュエータの振舞いや，システムの状態遷移に対してリアルタイム性に関する処理がプログラムの各所に散在する．このことからプログラムの変更可能性，可読性が低下し，保守しにくくなるという問題がある．

本研究の目的は，コンテキストとリアルタイム性の横断的関心事をアスペクトとして統一的に扱うソフトウェアアーキテクチャを設計することである．このアーキテクチャに基づくことで保守性の高い組み込みソフトウェアの開発を可能にすることである．

本研究室では，コンテキストウェアな組み込みシステムのためのアーキテクチャを提唱している [5]．これは，コンテキストをアスペクトとして分離している．一方，E-AoSAS++ [2] では，リアルタイム性をアスペクトとして分離している．これらを参考にし，コンテキストとリアルタイム性の関係を明らかにする．結果として，コンテキストとリアルタイム性をアスペクトとして分離して統一的に扱うアーキテクチャを設計する．本研究で設計するアーキテクチャの妥当性を確認するために，運転支援アプリケーションを事例とする．アーキテクチャに基づく実現と，一般的なオブジェクト指向での実現の比較を行い，アーキテクチャの妥当性について考察する．

2 背景技術

2.1 アスペクト指向技術

アスペクト指向技術は，横断的関心事 (cross cutting concern) をアスペクトとして独立してモジュール化する技術である [7]．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 コンテキスト指向技術

コンテキストに応じた振舞いを変化することができるようにモジュール化するプログラミング手法をコンテキスト指向プログラミング (Context Oriented Programming : COP [6]) と呼ぶ．コンテキストとはプログラムから観測することができる外部環境やシステムの内部状態で，時間や場所とともに変化し，それがプログラムの様々な処理の実行に影響を与えるものを指す．コンテキストに応じた振舞いをレイヤというモジュールに記述し，コンテキストに応じたレイヤを切り替える．

2.3 コンテキストウェアな組み込みシステムのためのアーキテクチャ

江坂らの提唱しているコンテキストウェアな組み込みシステムの構造を規定するアーキテクチャが図 1 である．本アーキテクチャは，オブジェクト指向をプライマリコンサーンとしている．また，並行性，実時間性，耐故障性などの非機能特性をセカンダリコンサーンとして設計している．セカンダリコンサーンの分離にアスペクト指向技術を適用することで実現している．

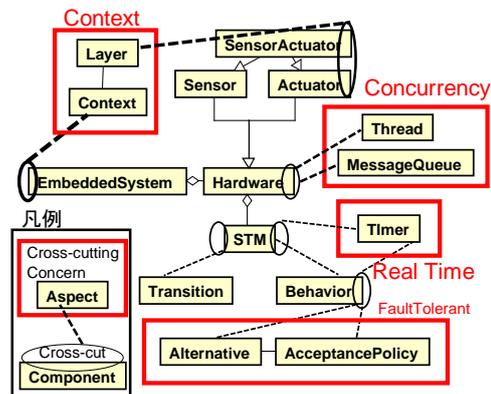


図 1 コンテキストウェアな組み込みシステムのためのアーキテクチャ

2.4 E-AoSAS++

E-AoSAS++ は本研究で提案している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．組み込みソフトウェアを並行に動作する並行状態遷移機械 (以下，CSTM) の集合として規定している．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，並行性，状態遷移，エラー処理，耐故障性，実時間性の 5 つを横断的関心事として定義している．

3 リアルタイム性を考慮したコンテキストウェアな組込みソフトウェアアーキテクチャ設計

コンテキストとリアルタイム性を考慮した場合，コンテキストに応じたアクチュエータの振舞いや，システムの状態遷移に対してリアルタイム性に関する処理がプログラムの各所に散在する．このことから組込みシステム的设计では，コンテキストとリアルタイム性が各所に横断している．我々はアスペクト指向技術を適用し，コンテキストとリアルタイム性をアスペクトとして分離することで，これらを統一的に扱う組込みソフトウェアアーキテクチャを提案する．

3.1 コンテキストとリアルタイム性の定義

本研究で扱うコンテキストは，センサが取得する物理量と対象システム全体を表現する概念 [3] とする．リアルタイム性は，あるオブジェクトに対して時間制約があり，またコンテキストの変化に対して時間制約が存在すると定義する．

3.2 組込みソフトウェアアーキテクチャの設計

本研究室で提案しているコンテキストウェアな組込みシステムのためのアーキテクチャに基づいてコンポーネントの構造を定義した．設計した組込みソフトウェアアーキテクチャを図 2 に示す．コンテキストコンサーンは，オブジェクト指向に基づく分割とは異なる役割を規定する．システムの状態に応じて振舞いに変化することから，Hardware 全体に横断する関心事である．EmbeddedSystem の内部状態を Context 複数のアクチュエータの振舞いを Layer として分割する．リアルタイム性コンサーンは，オブジェクト間のメッセージ通信をきっかけとして，リアルタイム性コンサーンに関する処理が実行される．アクチュエータの振舞いの制御では，一定時間内に処理を実行するために，時間を計測する必要がある．特定の処理の一部に，リアルタイム性に関する処理を与えることからコンテキストと STM の Behavior の一部に対してリアルタイム性コンサーンが横断する．その構造としては，オブジェクト指向と親和性が高いことから Timer によってこれを実現する [1] ．

3.3 システムの状態遷移

コンテキストに応じたレイヤの活性化を図 3 に示す．SensorActuator が物理量を計測し，コンテキストにイベント通知を行う．Context がコンテキストの状態を推定し，LayerActivator に更新通知をする．LayerActivator

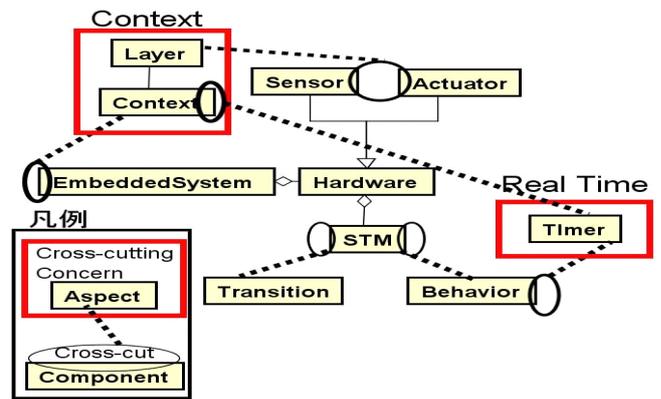


図 2 組込みソフトウェアアーキテクチャ

がコンテキストの状態に応じた Layer の活性化を行う．活性化された Layer がコンテキストの状態に応じたハードウェアの振舞いをアクチュエータにイベント通知を行う．

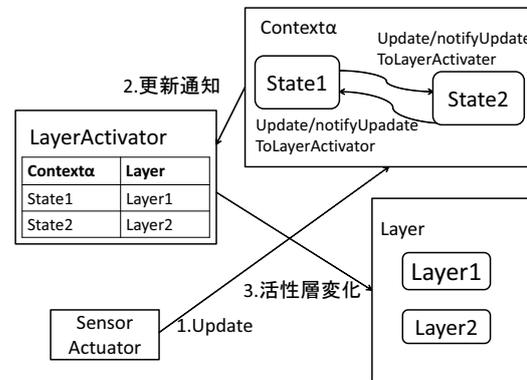


図 3 コンテキストに応じたレイヤの活性化

リアルタイム性を考慮したレイヤの状態遷移を図 4 に示す．レイヤの各処理は決められた一定時間内に動作しなければならない．アクチュエータが動作している時間を計測するために，タイマが独立してモジュール化する．RealTime_IAD はレイヤとタイマの間に定義した．Wait 状態から Activate 状態に遷移する．Act1 のアクションが行われたと同時に，Timer が Start する．決められた時間が経過すると Timer が Timeout し，Realtime_IAD がイベントとして Ev2 を通知する．

図 5 はコンテキストの状態遷移とリアルタイム性の関係を示している．コンテキストの状態が切り替わると同時に Timer を起動させる．指定されたデッドラインまで時間を計測する．デッドラインを越えてしまうと Timer が Timeout する．エラーと判断して起動中の Layer を非活性化し，例外処理に応じた Layer を活性化する．これにより，コンテキストとリアルタイム性の関係が明らかになった．

コンテキストに横断しているリアルタイム性を分離して独立にモジュール化することにより，互いに影響を及ぼすことがなくなった．その結果，変更が容易になる．

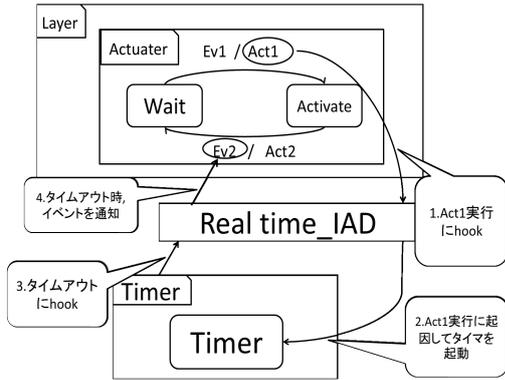


図 4 リアルタイム性を考慮したレイヤの状態遷移

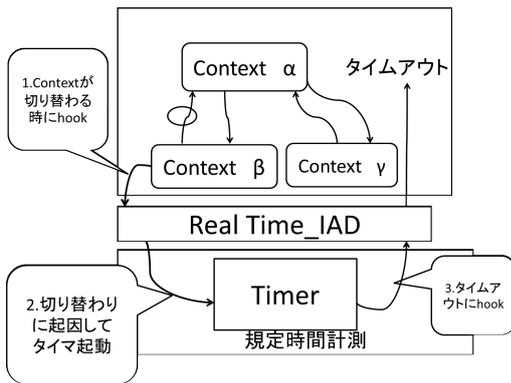


図 5 コンテキストとリアルタイム性の関係図

リングを左右方向に動作し直進する．超音波センサが検知した車両の速さに合わせて追従走行する．

4.2 事例のモデル化

適用事例を搭載した車両のハードウェア構成とリアルタイム性の関係を，我々のアーキテクチャに基づいて詳細化した．図 6 は適用する事例のモジュール構造である．アーキテクチャのコンテキストとレイヤを詳細化する．コンテキストを車間距離と路面状況と定義する．レイヤとして駆動モータ (ACT_M) 弱回転レイヤ，駆動モータ強回転レイヤ，ステアリングモータ (STEM) 左回転レイヤ，ステアリングモータ右回転レイヤ，Stop レイヤを定義した．

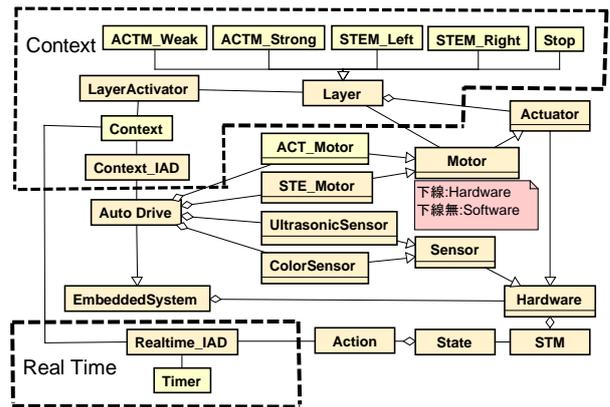


図 6 適用事例の詳細設計

4 事例検証

既に製品化されている運転支援アプリケーションの Adaptive Cruise Control(以下, ACC[8]), Lane Keeping Assist(以下, LKA[9]) を事例として, 設計したアスペクト指向アーキテクチャによる実現の妥当性を考察する．

4.1 適用する事例の仕様定義, シナリオ

適用する事例の仕様と基本動作を以下に示す．

- 前方車両を検知して追従走行 (ACC)
- 車線を検知しながらレーン内を走行 (LKA)

車両が後退することはないとする．追従走行するとき, 自車の加減速の上限と下限をあらかじめ一定の数値に設定する．例外処理としてレーンを見失った, あるいははずれた時間が長い時は動作を停止する．適用する事例を実装する 4 輪前輪駆動車両のハードウェア構成は, 車線の色を検知するために, 左右の前輪に 1 つ, 計 2 つの色彩センサを持つ．前方車両との車間を検知するために, 超音波センサを持つ．方向制御をするステアリングモータを一つ持つ．速度制御をする駆動モータを一つ持つ．

一連のシナリオとして, 以下を定義する．レーン内の中央から動作を開始する．色彩センサがレーンの色を検知する, または超音波センサが前方車両を検知するまで一定直進する．左右色彩センサがレーンの色を検知したらステア

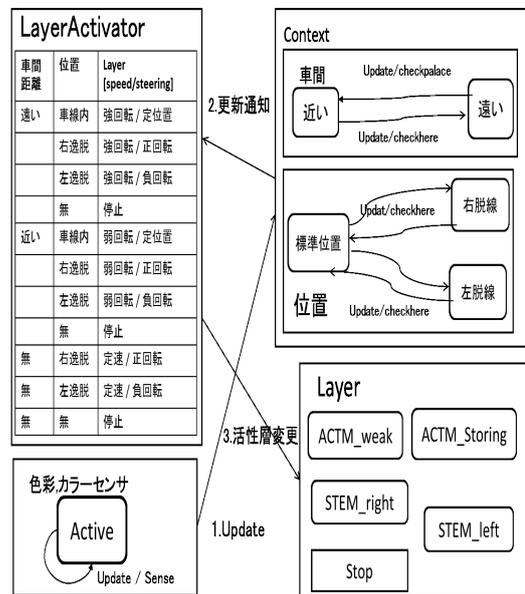


図 7 適用事例のコンテキストに応じたレイヤの活性化

適用する事例のコンテキストに応じたハードウェアの振舞いを図 7 に示す．LayerActivator が各コンテキストの組み合わせによってレイヤを活性化する．

リアルタイム性を考慮したレイヤの状態遷移の一例を図 8 に示す．コンテキストが「車間距離が近い」とき, 減

速 Layer が活性化される．活性化された減速 Layer が一定に回転している駆動モータに弱回転イベントを通知する．イベントを通知された駆動モータが弱回転開始を行う．弱回転が行われたと同時に Timer を Start させる．Timer が 5 秒になるまで計測し，計測完了つまり Timeout を RealTime_IAD に通知する．駆動モータに中回転を送信し，中回転開始を行う．

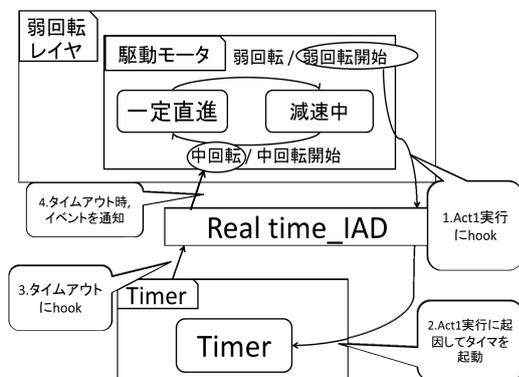


図 8 リアルタイム性を考慮した適用事例のレイヤの状態遷移

5 考察

事例をオブジェクト指向実現とアーキテクチャを適用した実現との比較をし，アーキテクチャの妥当性について考察する．図 9 では，左のコードがオブジェクト指向による実現，右のコードが我々のアーキテクチャによる実現である．オブジェクト指向による実現では，1 つのオブジェクトにすべてのコンテキストに対応する振舞いがメソッドとして実現され，それぞれのメソッドに対してリアルタイム性に関する記述が散在する．我々のアーキテクチャによる実現では，レイヤとリアルタイム性に関する処理をモジュール化し，それぞれの変更を独立に行うことが可能となっている．

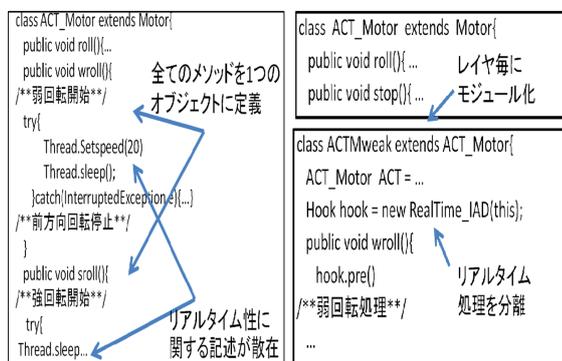


図 9 オブジェクト指向による実現 (左) とアーキテクチャを適用した実現 (右)

適用事例のコンテキスト時の処理は，それぞれ駆動モータの正方向回転速度とステアリングモータの左右方向回転

を行うメソッドとして定義されている．それぞれのメソッドに時間を計測して停止するリアルタイム性に関する処理が散在している．

本研究で提案するアーキテクチャは，コンテキストとリアルタイム性をアスペクトとして統一的に扱うことを可能にした．プログラムの変更の容易性や再利用性，また可読性が向上し，保守がし易くなる．

6 おわりに

本研究では，保守性の高い組込みソフトウェアの開発を可能にするために，コンテキストとリアルタイム性をアスペクトとして統一的に扱うソフトウェアアーキテクチャを設計した．運転支援アプリケーションを事例として，アーキテクチャに基づくことでそのコードの柔軟性が確保されることを確認した．今後の課題は，異なる具体的な事例に適用することでアーキテクチャの妥当性を確かめることが挙げられる．

参考文献

- [1] B. Douglass, “Real-time design patterns: robust scalable architecture for real-time systems,” *Addison-Wesley professional*, vol. 1, 2003.
- [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*, pp. 206–213, 2007.
- [3] 青山幹夫, 杉田達也, 中道上, “概念コンテキストモデルに基づくコンテキストウェアサービス開発手法の提案,” 情報処理学会第 37 回全国大会, 2011.
- [4] 一場利幸, “組込みリアルタイムシステムを対象としたマルチコアプラットフォーム技術,” 名古屋大学博士学位論文, pp.1–3, 2013.
- [5] 江坂篤侍, 野呂昌満, 沢田篤史, 繁田雅信, 谷口弘一, “コンテキストウェアネスを考慮した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の適用と実現,” *ソフトウェア工学の基礎ワークショップ (FOSE2016) 論文集*, vol.23, pp.175–180, 2016.
- [6] 紙名哲生, “文脈指向プログラミングの要素技術と展望,” *コンピュータソフトウェア*, vol.31, no.1, pp.3–13, 2014.
- [7] 千葉滋, “アスペクト指向ソフトウェア開発とそのツール,” *情報処理*, vol.45, no.1, pp.28–33, 2004.
- [8] JAF, “ACC(Adaptive Cruise Control),” http://jaf-acc.jp/about_acc/, 2016.
- [9] TOYOTA, “LKA(Lane Keeping Assist),” https://toyota.jp/information/campaign/anzen_anshin/gijyutu/06/, 2016.