

組み込みソフトウェアのソフトウェアアーキテクチャに関する研究 － 自動車自動制御を例題として －

2000MT033 石見 知也

指導教員 野呂 昌満

1 はじめに

近年, ITS(Intelligent Transport System) に代表されるような自動車の自律走行に関する研究が行われており, オブジェクト指向を用いたソフトウェアアーキテクチャが提案されている [2]. オブジェクト指向を用いるのは, 技術の進歩への対応が容易であることや, 使用者の要求拡大によって開発対象が大規模化しているという背景に基づいている.

オブジェクト指向で自律走行自動車ソフトウェアを実現したさいに, 実時間処理や同期処理が複数のオブジェクトと横断的に関係する構造になることを発見した. これはオブジェクト指向では, 複数のオブジェクトにまたがる大域的な特性をオブジェクトとして自然な形で取り扱うことが困難であることに起因する. 散在する処理は, ソフトウェアの柔軟性を低下させるので, 制御ソフトウェアの実現において, 解決しなければならない問題のひとつであると考えた.

本研究の目的は, アスペクト指向 [1] を用いた自律走行自動車ソフトウェアのアーキテクチャを提案することである. 他の研究では, ITS 関連の制御ソフトウェアの実現のために, オブジェクト指向を用いたアーキテクチャを提案している. 本研究では, オブジェクト指向を用いるとコンサーン [1] が複数のオブジェクトと横断的に関連するという問題に注目し, アスペクト指向を用いてその解決を試みる. コンサーンの分離を行い, 複数のオブジェクトに散在した処理を局所化することにより, 変更柔軟に対応できるアーキテクチャを構築できると考える.

提案したアーキテクチャに基づいて自律走行自動車ソフトウェアを実現し, その構造を考察するという手順で研究を行う. 考察では, ハードウェア変更・機能の追加などの例を用いて, 提案したアーキテクチャの柔軟性について議論する.

2 自律走行自動車ソフトウェアのアーキテクチャ

アスペクト指向を用いた自律走行自動車ソフトウェアのアーキテクチャを提案する. アーキテクチャは, オブジェクト間を横断するコンサーンを抽出し, アスペクト指向によって抽出したコンサーンの分離を行うことによって構築する. ハードウェア制御をコアコンサーンとした自律走行自動車ソフトウェアにおいて, 実時間処理, 同期処理, 情報提供に関する処理をコンサーンとして抽出した. ハードウェアの故障を検知するためには実

時間処理を行う必要があるが, ハードウェアを操作する側のオブジェクト内にタイマ起動などの処理が散在するので, コンサーンとして抽出した. 同期処理は, 複数のハードウェアから同時に要求が行われる可能性がある箇所すべてに記述する必要があり, オブジェクトとして自然な形でモジュール分割することが困難なので, コンサーンとして抽出した. 情報提供は, ディスプレイの更新を行うために, Observer パターンを用いて状態の変更を通知する構造にした場合の問題である. 状態変更通知を行う側のオブジェクトに, リスナーの追加, 削除, 変更通知などの処理が散在するので, コンサーンとして抽出した. 複数のオブジェクト間を横断する処理を局所化するために, アスペクト指向を用いてコンサーンの分離を行う. アスペクト指向を用いた自律走行自動車ソフトウェアのアーキテクチャを図 1 に示す.

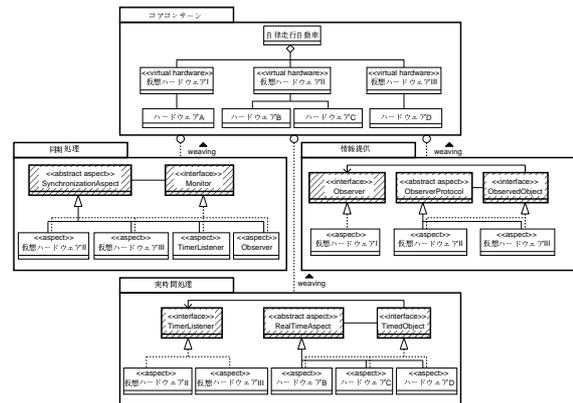


図 1 自律走行自動車ソフトウェアのアーキテクチャ

3 自律走行自動車ソフトウェアの実現

提案したアーキテクチャに基づいて, 自律走行自動車ソフトウェアの実現を行う. 実現を行うことにより, 提案したアーキテクチャの実用性を示す. 自律走行自動車への要求として, a) 運転する, b) 運転を中断する c) 現況を報告する, d) 経路探索する, を考えた. 運転の開始と中断という要求を抽出したのは, 自律走行を行うために最低限必要だからであり, 現況を報告するという要求を実現するのは, 走行開始と中断の判断を使用者が行うために走行中の情報が必要だからである. また, 交差点ごとに使用者が進行方向を選択する必要がないように, 経路探索を行う.

4 考察

実現した自律走行自動車ソフトウェアを用いて, 提案したアーキテクチャについて考察する. 具体的にハードウェア変更・機能の追加などの例を用いて, 提案したアーキテクチャの柔軟性について議論する.

ハードウェアの変更

実現した自律走行自動車の入出力機器をタッチパネルに変更した例を用いて、提案したアーキテクチャの柔軟性について議論する。実現した自律走行自動車は、ボタンとディスプレイを入出力に用いていた。ディスプレイは、相互排除を行う同期処理と、タイマを用いて故障を検知する実時間処理を行っている。一定時間内に終わらせる必要がある処理は、処理の開始直前にタイマを起動させ、処理が終了する前に規定時間を過ぎるとエラーを発生するようにしている。入出力機器をタッチパネルに変更する場合、ディスプレイに関する同期処理と、故障検知のための実時間処理も同時に書き換える必要がある。オブジェクト指向を用いて自律走行自動車ソフトウェアを実現した場合、実時間処理や同期処理は複数のオブジェクトと横断的に関連する構造となる。実時間処理が複数のオブジェクトと横断的に関連する構造を図2に示す。提案したアーキテクチャを用いると、ディスプレイの実時間に関する処理を分離して記述できる。ディスプレイのメソッドを呼び出す側のオブジェクトには、実時間に関する処理を記述することがないので、タッチパネルへの変更は、ディスプレイのメソッド呼び出し箇所と、実時間処理パッケージ内のディスプレイアスペクトを書き換えるだけで変更可能である。

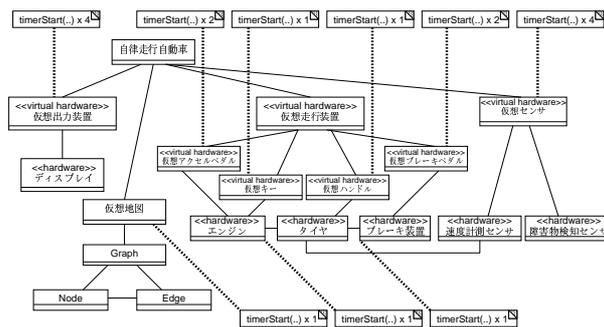


図2 複数のオブジェクト間を横断する実時間処理

ハードウェアの機能追加

実現した自律走行自動車のディスプレイに、省電力機能を追加した例を用いて、提案したアーキテクチャの柔軟性について議論する。ここでいう省電力機能とは、ディスプレイへの更新要求が一定時間以上行われなかった場合、使用していないものと判断して画面を非表示にする機能のことを表す。本研究で提案したアーキテクチャでは、アスペクト指向を用いることにより、実時間処理を分離して記述することができる。一定時間内に終わらせる必要がある処理や、一定時間ごとに繰り返す処理は、ジョインポイントとしてフックすることにより実現可能である。省電力機能を追加する場合、省電力モードに移行するメソッドを追加し、そのメソッドを一定時間経過後に呼び出せばよい。省電力機能を追加するための記述を図3に示す。このディスプレイの省電力機能の処理は、実時間処理パッケージ内に局所化されているので、さらに別のハードウェアに省電力機能を追加する場合も容易に流用できる。

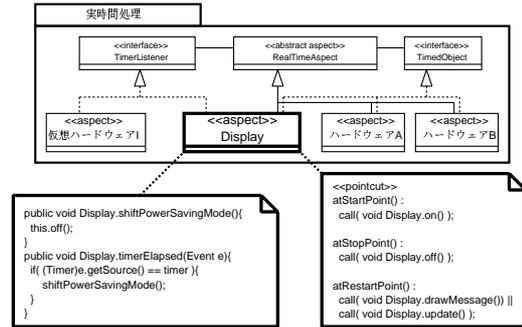


図3 省電力機能追加のための記述

実現過程におけるコンサーンの分離

分析・設計段階で想定していなかったコンサーンを、実現過程で新たに抽出した場合の例を挙げ、提案したアーキテクチャの柔軟性について議論する。作成した自律走行自動車ソフトウェアでは、故障検知のために実時間処理を行っているが、タイマ起動直前に情報を記録するように変更した場合を考える。実時間処理を分離せずに記述している場合、故障検知を行っている箇所をすべて特定し、タイマ呼び出しの前に情報記録のための処理を記述する必要がある。故障検知のための実時間処理は、複数のオブジェクトと横断的に関連しており(図2)、それらすべてを変更するのは困難である。提案したアーキテクチャでは、実時間に関する処理は、実時間処理パッケージに局所化されており、タイマを起動する箇所は容易に特定できる。実時間処理パッケージ内のタイマ起動箇所をジョインポイントとするロギングパッケージを、提案したアーキテクチャに追加すればよい。

5 おわりに

本研究では、アスペクト指向を用いた自律走行自動車ソフトウェアのアーキテクチャを提案し、ハードウェア変更・機能追加の例を用いて、その柔軟性について議論した。他の研究では、オブジェクト指向を用いたITS関連の制御ソフトウェアの実現を提案しているが、コンサーンが複数のオブジェクトと横断的に関連するという問題に注目し、アスペクト指向を用いて問題を解決した。提案したアーキテクチャの抽象化を行い、他のITS関連の制御ソフトウェアに適用することを今後の課題とする。謝辞

本研究を進めるにあたり、二年間にわたって御指導頂いた野呂昌満教授、有益なアドバイスや指摘を頂いた張漢明助教授、蜂巣吉成講師、熊崎敦司さん、大学院生の藤原泰昌さん、森貴彦さん、後藤修平さんに深く感謝します。

参考文献

- [1] T. Elrad, R. Filman, A. Bader, Eds.: Special Issue on Aspect Oriented Programming, CACM, Vol.44, No.10, pp.29-32 (2001).
- [2] 後藤俊蔵, 堀豊: JSK 通信 第8回 ITS 世界会議特集号, 自動車走行電子技術協会, pp.1-10 (2001).
- [3] 松下 温, 屋代 智之: ITS の実現に向けて, 情報処理, Vol.40, No.10, pp.960-963 (1999).