

# 自動車組込みシステムのサービス指向開発に関する研究

2002MT067 小木曾 裕規 2002MT068 大勝 大輔

指導教員 青山 幹雄

## 1. はじめに

自動車組込みソフトウェアシステム(以下、自動車組込みシステム)は、ソフトウェアコンポーネントを使用して開発している。しかし、ソフトウェアコンポーネントでの開発にも動作環境が決まっているなどの問題がある。本研究では、この問題の一解決方法としてサービス指向の自動車組込みシステム開発を提案する。

## 2. ソフトウェアコンポーネント

### 2.1. ソフトウェアコンポーネントの特徴

コンポーネント同士が、あらかじめ規定したインターフェースを介して結合される。相互依存性を低減でき、コンポーネントの独立性が向上する。これにより呼び出し関係の整合性を保証したコンポーネントの再利用が可能となる[6]。

### 2.2. ソフトウェアコンポーネントの問題点

#### (1) システムが密結合になる

インターフェースがバイナリデータを介した遠隔手続き呼び出しの接続になる。バイナリデータの場合、プラットフォームや開発言語に依存しやすく、アプリケーション同士の結合度が強くなる。

#### (2) コンポーネントは動作環境が決まっている

ベンダーは様々な環境を使用して開発しているため、開発の最終段階で 1 つのシステムとして統合したときに不具合が生じるなどの問題がある。これによりコストの増加や開発期間の増大につながる。

## 3. サービス指向アーキテクチャ(SOA)

### 3.1. サービス指向アーキテクチャ(SOA)とは

SOA(Service-Oriented Architecture)とは、大規模システムをサービスの集まりとして構築する設計手法である。サービスとは、外部から標準化された手順によって呼び出される 1 まとまりのソフトウェアであり、意味のある機能を持つもの。

### 3.2. 自動車組込みシステム開発で SOA を使うねらい

#### 3.2.1. SOA の特徴

SOA を使うねらいを SOA の特徴から示す。

#### (1) サービスは疎結合である

テキストデータである XML を利用した接続技術になる。この接続技術が SOAP になる。テキストデータ形式なので、エディタで開けば簡単にデータが含まれているのか確認

でき、タグによってデータにどんな属性が付けられるのかも理解しやすくなる。また、データ変換も容易になる。

#### (2) インタフェースが特定のプラットフォームに依存しない

多くのベンダーが様々なプラットフォームを使用して開発しても、プラットフォームと独立になる。また、設計された時代、設計当初の用途などを超えて、サービス同士の通信が可能となる。

### 3.2.2. SOA の導入

SOA の特徴から、SOA を導入することによって自動車組込みシステム開発におけるソフトウェアコンポーネントの問題が解決できるといえる。よって本研究では、SOA を導入した自動車組込みシステム開発を提案する。

## 4. サービス指向開発の流れ

図1にサービス指向開発の流れを示す。

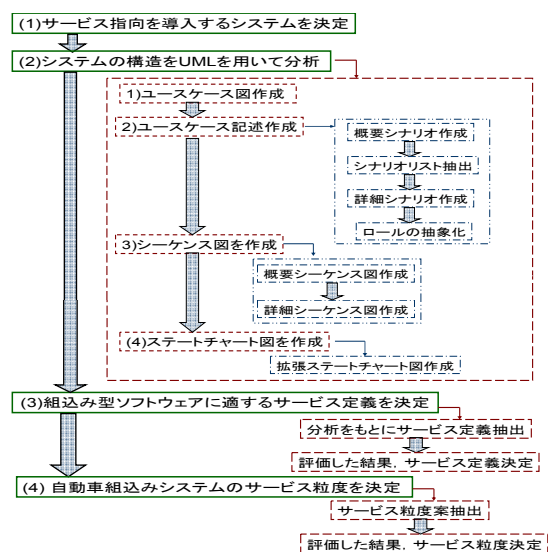


図1 自動車組込みシステム開発の流れ

## 5. クルーズコントロールシステムの分析

### 5.1. クルーズコントロールとは

クルーズコントロールとは、設定速度で自動定速走行する速度制御装置である。

具体的にサービス定義、粒度を抽出するために、例としてクルーズコントロールを用いて分析する[2]。

### 5.2. ユースケース図

クルーズコントロールのアクタは、ドライバ、シャフト、タイマ、スロットルとなる。多くのユースケースをグループ化し、図2でユースケースのパッケージ図として示す[5]。

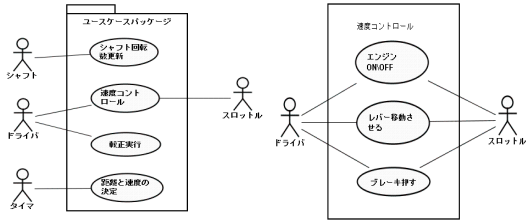


図2 クルーズコントロールのユースケースパッケージ図

図2のユースケースパッケージ中で、クルーズコントロールの主要部分である「速度コントロール」に着目する。

我々は、この「速度コントロール」をサービスとして定義できないかと考え、分析した。図3は「速度コントロール」のユースケース図である。

### 5.3. ユースケース記述

ユースケース記述で、シナリオを概要と詳細に分けて考えた。このように分けることで、分析段階でサービス粒度の見通しを立てる。概要シナリオはまとまった機能に対する振る舞いで、詳細シナリオはシステム構成要素単位の振る舞いを示す。

|  |
|--|
| ユースケース名: 速度コントロール  |
| アクター: ドライバ, スロットル  |
| 要約: ドライバが、クルーズコントロールレバー、ブレーキ、エンジン外部入力装置を使って入力を与えることにより行われるオートクルーズコントロールを記述 |
| メインフロー:<br>1. ドライバはアクセル位置にレバーを移動し、そのままの状態にする。システムは自動加速を始める。...             |

図4 「速度コントロール」の概要シナリオ記述

図4の概要シナリオ記述だと、レバーを各位置に移動させたときのシステムの構成要素の部分不明確なので、さらに詳細に分析する。レバーを各位置に移動させたときのシナリオリストとして、システムの構成要素の振る舞いを表したものを詳細シナリオ記述とした。

|  |
|--|
| シナリオリスト: レバーをアクセル位置に移動、レバーをセット位置に移動、レバーをレジューム位置に移動、ブレーキを押す、レバーをオフ位置に移動 |
| シナリオ記述:<br>・レバーをアクセル位置に移動させた場合...                                      |

図5 「速度コントロール」の詳細シナリオ記述

### 5.4. シーケンス図とステートチャート図を作成

シーケンス図を概要と詳細の2段階に分けて作成した。

概要シーケンス図はサービス単位の振る舞いを表す。詳細シーケンス図は、サービス単位内の振る舞いを表す。

図6の詳細シーケンス図はシナリオリストにおける各場合のシーケンス図を詳細シーケンス図とする。

#### (1) 概要シーケンス図

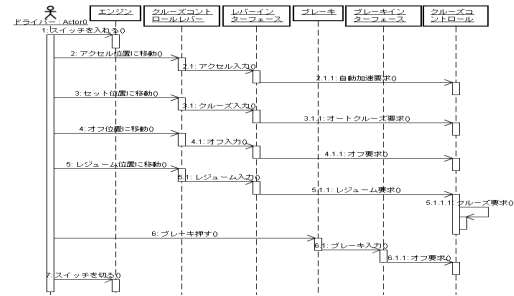


図6 「速度コントロール」のシーケンス図

#### (2) 詳細シーケンス図

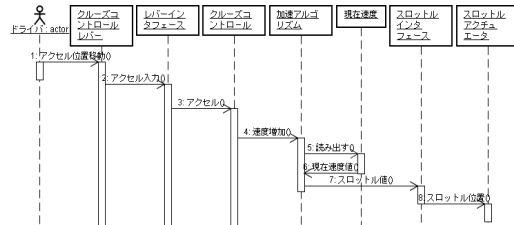


図7 レバーをアクセル位置に移動のシーケンス図...

#### (3) ステートチャート図

概要と詳細シーケンス図を基に、取りうる状態を以下のように抽出し、ステートチャート図を作成する。

自動車が動いていない ⇒ 停止状態、 エンジンスイッチオン、自動車が動いている ⇒ 初期状態、 システムが自動加速する ⇒ 加速状態、 オートクルージングする ⇒ クルージング状態、 クルーズコントロールオフにする ⇒ クルージングオフ状態、 オートクルージングを再開する ⇒ レジューム状態

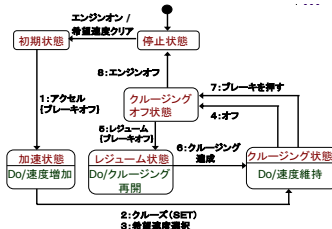


図8 「速度コントロール」のステートチャート図

構造を分析した結果、次のことが明らかになった。

- 1) ユースケースとシーケンス図を概要と詳細の2段階に分析したことで、サービス分析の見通しがついた。
- 2) ステートチャート図でスタートからストップまで1つの機能を表していることが分かった。
- 3) ステートチャート図でセンサ信号からアクチュエータまで状態遷移の一連の流れが分かった。
- 4) メインフローを検査し、例外フローも分析しなおし、ステ...

トチャート図も見直さなければならない。

この4点を満たすステートチャート図を、次のような拡張ステートチャート図として作成した。

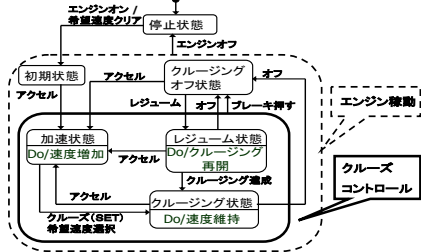


図9 「速度コントロール」の拡張ステートチャート図

これにより、クルーズコントロールによる自動運転の範囲と手動運転の範囲を分けることで、自動運転がサービスの定義、粒度を決めていく手がかりなどが得られた。

## 6. サービス定義決定

### 6.1. 分析をもとにサービス定義を抽出

分析をもとにサービス定義を抽出する。最終的に、自動車組込みシステムでのサービス定義を決定する[3, 4]。

ユースケース記述やシーケンス図などから、クルーズやレジャー等の独立性のある機能単位のまとまりが多数あることが分かった。そこで、サービス定義を次のようにする。

**サービス定義案{1}: 意味のある機能を提供しているもの**  
上記のサービス定義例を図10に示す。

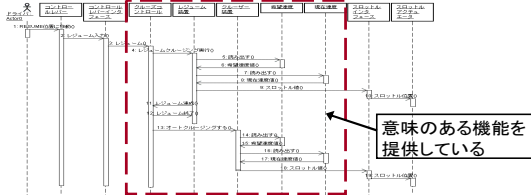


図10 レジャーのシーケンス図

### 6.2. 抽出したサービス定義を評価

#### (1)異なる方法基準のサービス定義を抽出

サービス定義を別の観点から考えるために文献[4]を用いる。この文献[4]から、次のサービス定義を抽出した。

**サービス定義案{2}: エンティティ間に及ぼす作用**

上記のサービス定義の例を図11に示す。

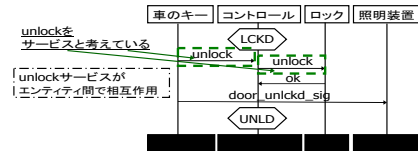


図11 エンティティ間の関係

#### (2)サービス定義を評価

サービス定義案{1}, {2}を評価し、サービス定義を決定する。サービス定義案{2}の考え方は、多くのコンポーネントがまとまっているサービスの中から unlock サービスを呼び

出している。言い換えれば、独立性のある機能単位のまとまりであるサービスから unlock サービスを呼び出す。つまり、サービス定義案{1}を包含している。以上のことから本研究でのサービス定義をサービス定義案{1}に決定する。

### 6.3. サービス定義決定

組込み型ソフトウェアでのサービス定義を決定した。

**サービス定義: 意味のある機能単位**

このサービス定義をもとに、以下でクルーズコントロールのサービス粒度を決める。

## 7. サービス粒度決定

### 7.1. 分析をもとにサービス粒度案を抽出

拡張ステートチャート図からサービス粒度案を抽出した。赤の実線の各状態は、意味のある機能単位でまとまりがあるので、サービス定義に該当する。緑二重線の範囲が示すクルーズコントロール全体が、粒度の最大範囲。

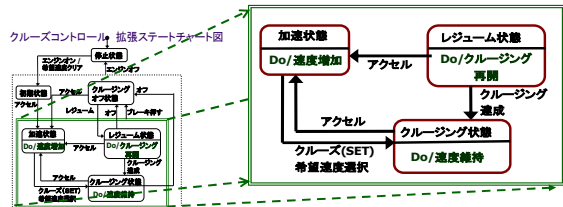


図12 分析からのサービス粒度案

### 7.2. 異なる仕様からサービス粒度案抽出

抽出したサービス粒度案を比較・評価するために、他のクルーズコントロール[1]からサービス粒度案として、クルーズアルゴリズムを抽出した。クルーズアルゴリズムでは、図12の赤の実線部分のセットスピードが主となって速度調整をしている。このセットスピードをサービス粒度案とする。この案は、サービス定義と一致する[1]。

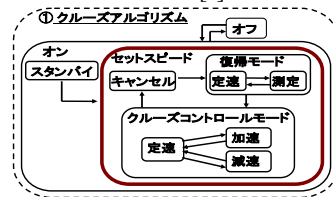


図13 クルーズアルゴリズムとサービス粒度案

### 7.3. 抽出したサービス粒度案を比較

二つのサービス粒度案のステートチャート図を図14に示す。

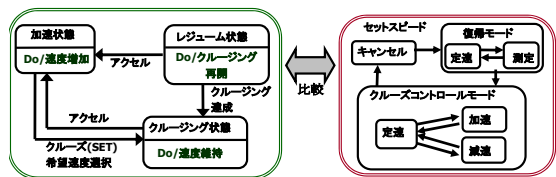


図14 サービス粒度案の比較

比較の結果、セットスピードをサービス粒度として考える。この理由は、図 15 に示すように、共通して使用され再利用性があるからである。

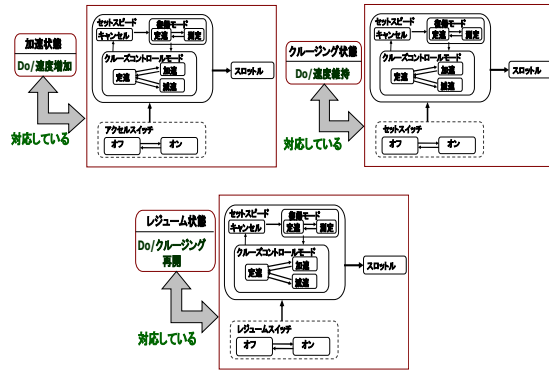


図 15 各状態とクルーズアルゴリズムの対応

#### 7.4. 抽出したサービス粒度を評価

評価基準(1), (2)を定め、サービス粒度案を評価する。

評価基準(1): 意味のあるまとまりとする

評価基準(2): 制御ロジックを隠ぺいする

クルーズアルゴリズムの復帰モードは条件、制限部分などの車種ごとに異なるシステムを含んでいるので、評価基準(1)を満たさず、サービスに含めない。そこでサービスでない復帰モードと関係性があるロジックのキャンセルもサービスとして考えない。

残るクルーズコントロールモードは、共通する速度調整部分をサービスとしており、条件部分である復帰モードなどの他の機能が含まれていない。また、必要最小限のサービス粒度でありながら、どんなクルーズコントロールにも網羅できる。以上の点から、評価基準(1)を満たす。さらに、クルーズコントロールモードは、クルーズコントロールのアルゴリズムの部分なので制御ロジックは隠ぺいされ評価基準(2)も満たす。よってクルーズコントロールモードは、評価基準(1)と(2)を満たすので、クルーズコントロールのサービスとするのは適切である。

#### 7.5. サービス粒度決定

サービス粒度を以下のように決定した。

クルーズアルゴリズムのクルーズコントロールモード

### 8. 考察と評価

自動車組込みシステムは、ソフトウェアコンポーネントを使用して開発している。しかし、この開発にも問題はあ。本研究では、ソフトウェアコンポーネント開発の問題を解決すべくサービス指向に基づく自動車組込みソフトウェアシステム開発の方法を提案した。この方法により、サービスが疎結合の特徴を持つので、コンポーネントの多機能・高機能化の要求にも独立性の高状態が可能となる。さらに、インタ

フェースが特定のプラットフォームに依存しないので、再利用性の向上につながり、コスト削減や開発期間短縮が可能となる。サービス指向の導入によって、自動車組込みソフトウェアの大規模化・複雑化が緩和できると考える。

サービス定義案、サービス粒度案を比較・評価した結果から、サービス定義、サービス粒度を決定した。我々が定めた粒度がサービス定義を満たし、必要最小限で独立性や再利用性があるため、妥当なサービスであると考え。

### 9. 今後の課題

我々が提案したクルーズコントロールでのサービスを、他のシステムで使用し、再検討・再評価をする。これによりサービス粒度の評価の信頼度が向上し、より適切なサービス粒度が決まると考えられる。例えば、アダプティブ・クルーズ・コントロールなどにサービスを取り入れて、検討する。

今回決めた自動車組込みシステムでのサービス定義を他の組込みシステムに適応させ検証する。これにより自動車組込みシステムでのサービス定義の信頼性が向上する。

### 10. まとめ

本研究では、ソフトウェアコンポーネントを用いる自動車組込みシステム開発の問題解決として自動車組込みシステム開発へサービス指向開発を導入する方法を検討した。クルーズコントロールを事例にあげ、基本的技術として、サービス定義・粒度の基準を定めた。クルーズコントロールの構造分析から評価し、サービス定義・粒度を決定する方法を提案した。

### 参考文献

- [1] Cybernet Systems, Simulink/Stateflow サンプルモデル解説書—クルーズコントロール制御編, 2004.
- [2] H. Gomaa, Designing Concurrent, Distributed, and Real-Time Applications with UML, Addison Wesley, 2000.
- [3] I. H. Krüger, et al, Towards a Process and Tool-Chain for Service-Oriented Automotive Software Engineering, University of California, 2004.
- [4] S. Rittmann, Exploring Service-Oriented Software Development for Automotive Systems, Diploma Thesis, 2004.
- [5] 井上 樹, UML 徹底活用, 翔泳社, 2005.
- [6] 米持 幸寿, 基礎からわかる SOA, 日経BP社, 2005.