

車載向けソフト仕様書記述方式の研究と有効性実証
**Research on Specification Methodology of In-Vehicle Software
and its Practical Evaluation**

上野 高元¹、松本 倫子²、吉田 紀彦^{3 *1}
Takamoto Ueno¹, Noriko Matsumoto², Norihiko Yashida³

¹ 埼玉大学 工学部情報システム工学科
Department of Information and Computer Sciences, Saitama University

² 埼玉大学 大学院理工学研究科
Graduate School of Science and Engineering, Saitama University

³ 埼玉大学 情報メディア基盤センター
Information Technology Center, Saitama University

Abstract

This article presents a summary of joint research with Calsonic Kansei Corporation on the topic of specification methodology of in-vehicle software. Industrial embedded software, including in-vehicle software in particular, has a large variety of implementations under a single specification according to regional regulations, product grades, customer options and more. This research is to formalize each variation as a component which can be applied to a specification independently with each other. Using a modeling framework, MATLAB/Simulink, and a software engineering technique called “aspect oriented software”, we proved the ability of our approach on a practical in-vehicle software, “Auto Light System” provided by Calsonic Kansei Corporation.

Keywords: Embedded systems, System specification, System variation, Aspect-oriented software, MATLAB/Simulink

カルソニックカンセイ株式会社との間で平成 21 年春に開始した共同研究について、背景、目的、内容、平成 21 年度の成果をまとめる。なお、本共同研究は平成 22 年度も継続が予定されている。

1. 背景

組込みシステムとは、各種の機械や機器に組込まれ、その制御を行うコンピュータシステムである。その適用範囲は多岐に渡っており、近年の情報通信技術の発展により、組込みシステムは大規模化、複雑化してきている。

組込みシステムには、メモリ制約、CPU のクロック数制約などのリソース制約、高い信頼性、リアルタイム性など、厳しい要求が課され、その設計効率化を目指すためにシステムレベル設計 (System Level Design)、モデル駆動アーキテクチャ (Model-Driven Architecture)、モデルベース開発 (Model-Based Development) 等、様々な設計方法論が提案されてきた。

しかし、実際の開発現場ではシステムに課される要求はそれだけに留まらない。輸出先の環境の違い、部品の製造メーカーの違い、ユーザの好みの違い等により、ある特定の用途のシステムでありながら、様々なバリエーションのあるシステムが発生している。

*1 〒338-8570 さいたま市桜区下大久保 255
電話 048-858-3487, FAX 048-858-3716
Email yoshida@mail.saitama-u.ac.jp

2. 目的

例えば、ある一つの目的を果たすためのシステムを製造しようとしても、輸出先により法規、言語、習慣、環境が異なり、同じ用途のシステムでありながら異なるバリエーションが発生する。また、そのシステムを製造するために使用する部品も、メーカーによりそのシステムの一部を変えなければならないことがしばしば起こる。また、多くの分野ではその製品を使うユーザの好みによって異なる要求があり、システムのバリエーションが発生する原因となっている。

このように、ある一つの共通した目的に専用化して設計、開発されるシステムでありながら、国の環境、メーカーの違い、ユーザの好みなどの違いにより、異なるバリエーションが発生する。その具体例として、共同研究を行っているカルソニックカンセイ社において製造されている車載 ECU (Electronic Control Unit) について触れる。車載 ECU とは、自動車用制御コンピュータのことであり、マイクロプロセッサと周辺デバイス、通信モジュールによって構成される。車載 ECU が現れた当初はユニット単体で制御を行っていたが、最近では複数の ECU が協調して一つの、あるいは複数の機能を実現するようになった。

カルソニックカンセイ社では、様々な要求を満足するためにバリエーションに関わる機能を複数作成したが、それらを開発時、あるいは工場において選択・切替えを行ってシステムを製造していった結果、機能の増加と共に多数のバリエーションが作成されてしまい、その規模はもはや無視できない程まで膨れ上がってしまっている。

現場での対応の現状

バリエーション数に対応する最も単純な方法は、必要・不要を問わずにすべてのバリエーションに関わる機能を組み込むことである。そして、実際にそのシステムの実装バリエーションを作成する際には、工場や販売拠点で本当に必要な機能を選択する。しかしながら、この方法は機能毎の整合性を保つことが困難である。同時に、バリエーション数が増加すればすべての組合せを網羅した試験が不可能

となることから、動作保証が難しくなる。さらに、大量の無駄が発生してしまっている。

バリエーションに関わる部分がソースコードのある個所にまとまって現れ、異なるバリエーション同士がまったく独立しているのであれば、バリエーションのあるシステムを設計することは容易である。何故ならば、バリエーションに関わる機能をライブラリしてまとめることで部品化でき、システムの作成時に必要な機能を追加していけばよい。しかしながら、バリエーションに関わる機能はソースコード上の至る個所に散在し、異なるバリエーション間でも複雑に絡み合っている。

したがって、バリエーションのあるシステムを効率的に設計するためには、バリエーションに関わる各機能をシステム本体から分離することが課題となる。これまで述べた理由から、機能の分離をソースコードレベルで行うことには限界がある。すなわち、ソースコードレベルより抽象度の高いモデルレベルで機能の分離を行う必要があるが、現状ではその方法は考案されておらず、結局のところ新しいバリエーションが発生するたびにゼロからの設計を余儀なくされ、設計・検証のコスト増大、長期化を招いている。

望ましい対応

バリエーションを実装したシステムを開発する方法として、システムの開発をバリエーションに捉われない共通仕様から出発することが考えられる。バリエーションに関わる機能はシステムから分離させ部品化する。そして、システムの開発時には部品化したバリエーション機能を共通仕様に追加していく。しかしながら、その部品の作成には注意しなければならない。その理由と、部品の作成時に注意すべき事項を次に述べる。

バリエーションに関わらない共通仕様に関わる部品を組み込んでいく際に、部品は組み込む順序に捉われるべきではない。なぜならば、それが保障されないとき、異なる順序で部品を組み込んでいった結果、部品の追加順序により異なるシステムが作成されてしまう、もしくはそもそも部品の追加が不可能となってしまう可能性がある。

それにより同一のバリエーション機能に対して異なる部品を作成しなければならず、バリエーションのための部品をやみくもに増加させる原因となる。従って、理想的な部品とは、どのような順序で部品を組み込んだとしても、同じ部品を選択し、組み込んだのであれば最終的に作成されるシステムは同一のものとなることである。

以上のことから、増加するバリエーションに対し、バリエーションの為の部品の相互独立性を確保することが課題となる。

3. 内容

具体的な方針は、まず対象となるシステムをバリエーションに関わる機能を分離し、システム共通部分をバリエーションに捉われない共通仕様モデルとして作成する。一方、バリエーションに関わる機能は共通仕様から分離した部品として用意する。共通仕様モデルにバリエーション部分を組み込むことにより実装バリエーションモデルへ変換するが、変換するための部品の作成方法はアスペクト指向技術を利用する。つまり、分離したバリエーション部分をアスペクトとして部品化し、変換ソフトを作成することにより自動的に共通仕様に機能追加する。さらに、輸出先、メーカー、ユーザ等、様々なバリエーションの軸が考えられ、それらを実装するための部品を作成しなければならないが、それぞれのバリエーションの軸に関連する部品は互いに変換の順序を考える必要がない相互独立性を有しているものを目指す。最後に、変換過程で得られるすべてのモデルはシミュレーションによる検証が可能であるものとする。

これにより、共通仕様から実装バリエーションモデルへの変換は、要求されたバリエーションに変換するための部品の選択のみで十分である。さらに、各段階を検証・シミュレーション可能とすることで、早期の検証が可能となることから、設計の効率化が期待できる。

バリエーションのあるシステムを効率よく設計するためにはバリエーション部分とシステム共通部分を分離しなければならない。そしてそれはソース

コードレベルでは不可能であることから、より抽象度の高いモデルに対して行う必要がある。

バリエーションのあるシステムに対する設計方法の第一段階として、既存のシステムからバリエーションに捉われないシステムの共通仕様モデルを作成する。モデル記述は制御系業界で広く使用されている MATLAB/Simulink を用いる。MATLAB/Simulink はモデルベース開発に適したツールであり、システムのシミュレーション・検証が可能であることから効率のよい設計が可能である。

次の段階として、バリエーションに関連する機能を部品化しなければならない。その部品化の方法を、本研究ではアスペクト指向に基づいて行う。アスペクト指向の特徴の一つとしてクラス定義・関数定義のほぼ任意の個所にプログラム断片を追加することができ、そのプログラム断片だけをモジュール化・部品化出来ることがある。このアスペクト指向の特徴を共通仕様モデルから実装バリエーションモデルを作成するために利用する。これにより、システム共通部分と機能バリエーション部分の分離、および共通仕様への機能追加が可能である。

ここで、バリエーションに関わる部品には大きく分けて 2 つに分類することが出来ることを述べておく。一方は、その機能を追加した結果、モデルがそれほど変化しない部品、もう一方はその機能を追加した結果、モデルが大きく変化する部品である。モデルを大きく変化させない部品としては値・関数の追加、削除、変更等であり、モデルを大きく変化させる部品としては、モジュールの追加が代表的である。

本研究の題材として、共同研究を行っているカルソニックカンセイ社から御提供頂いた Auto Light System を扱う。Auto Light System とは、車載 ECU である BCM (Body Control Unit) の 1 つであり、外光に応じてヘッドランプ、テイルランプの制御、およびメーター明度調節を行う。Auto Light System には、以下の 3 つの軸によるバリエーションが存在する。

- 光学センサ値の正規化関数

- 地域特有部分
- 光学センサ値に対する閾値の決定

4. 成果

本研究では、共通仕様モデルからアスペクト指向を利用することにより様々な機能バリエーションを実装したモデルを作成する方法を提案した。その際、機能を追加するための部品はアスペクト指向に基づき作成し、異なるバリエーション間でも順序に拘わらずに機能追加を行っていくことが可能であることを実証した。

以下に今後の課題を挙げる。

アスペクト記述とモデルコンパイラの作成

本研究のモデル変換は手作業により行っているが、将来は自動化されることが望ましい。そのためには、MATLAB/Simulink モデルに対するアスペクト記述、および共通仕様モデルにアスペクトを適用するための変換ソフトを作成しなければならない。

より複雑なシステムへの設計方法の適用

モジュール同士が包含関係にあるような、より複雑なシステムに対して設計方法論を適用し、本研究で提案した設計方法論が真に有効であるかどうかを検証しなければならない。

関連主要業績

学術論文誌

- [1] N. A. Zakaria, N. Matsumoto, N. Yoshida, “Refactoring-based Executable UML Transformation for Embedded System Design”, *Int. J. Comp. Sci. Net. Sec.*, 9:6, 173–181 (2009)
- [2] M. E. Haque, N. Matsumoto, N. Yoshida, “Exploiting Context-Awareness in Cluster-Based Wireless Sensor Networks”, *IEEJ Trans. Elec. Electr. Eng.*, 4:5, 677–679 (2009)
- [3] M. Sugai, A. Teruya, E. Iwata, N. Matsumoto, N. Yoshida, “Design and Implementaiton of Executable UML Platform for Assertion-Based Dynamic Verification”, *Int. J. Comp. Sc.*, 3:5, 554–567 (2009)

[4] M. E. Haque, N. Matsumoto, N. Yoshida, “Context-Aware Cluster-Based Hierarchical Protocol for Wireless Sensor Networks”, *Int. J. Ad Hoc Ubiq. Comp.*, 4:6, 379–386 (2009)

[5] K. Shimizu, T. Sawamura, M. E. Haque, N. Matsumoto, N. Yoshida, “Dynamic Load-Aware Multicast Routing in Ad-hoc Networks”, *IEEJ Trans. Elec. Electr. Eng.*, 5:1, 123–125 (2010)

[6] M. E. Haque, N. Matsumoto, N. Yoshida, “Utilizing Multilayer Hierarchical Structure in Context Aware Routing Protocol for Wireless Sensor Networks”, *Int. J. Comp. Sci.*, 4:1, 23–37 (2010)

国際会議（全文査読）

[7] M. E. Haque, N. Matsumoto, N. Yoshida, “Context-Aware Multilayer Hierarchical Protocol for Wireless Sensor Network”, *Proc. 3rd Int. Conf. Sensor Tech. App.*, 277–283 (2009)

[8] T. Hoshino, A. Teruya, E. Iwata, M. Sugai, N. Matsumoto, N. Yoshida, “Communication Model Exploration in Aspect-Oriented Executable UML”, *Proc. Int. Conf. App. Comp. 2009, Vol.2*, 129–134 (2009)

[9] S. Kosuge, A. Teruya, E. Iwata, M. Sugai, N. Matsumoto, N. Yoshida, “Design Pattern Specifications in Aspect-Oriented Executable UML”, *Proc. Int. Conf. App. Comp. 2009, Vol.2*, 139–144 (2009)

[10] R. Suzuki, K. Shimizu, K. Kimura, C. Tsumura, M. E. Haque, N. Matsumoto, N. Yoshida, “Reliable and Geography-Aware Peer-to-Peer Multicast for Earthquake Early Warnings”, *Proc. Int. Conf. App. Comp. 2009, Vol.2*, 165–170 (2009)

以上の他、3編

専門誌記事

[11] 佐藤, 山郷, 高瀬, 鈴木, “ライン・トレース・カーで学ぶ組み込みシステム開発の基礎知識”, *Interface, CQ 出版社*, 130–141 (2009)