

# 車両情報制御システムの ソフトウェア開発プラットフォーム“PLATINA”

辰巳尚吾\*  
浅井陽介\*  
渡邊亮一\*\*

“PLATINA” : Software Platform for Train Integrated Management System

Shogo Tatsumi, Yosuke Asai, Ryoichi Watanabe

## 要旨

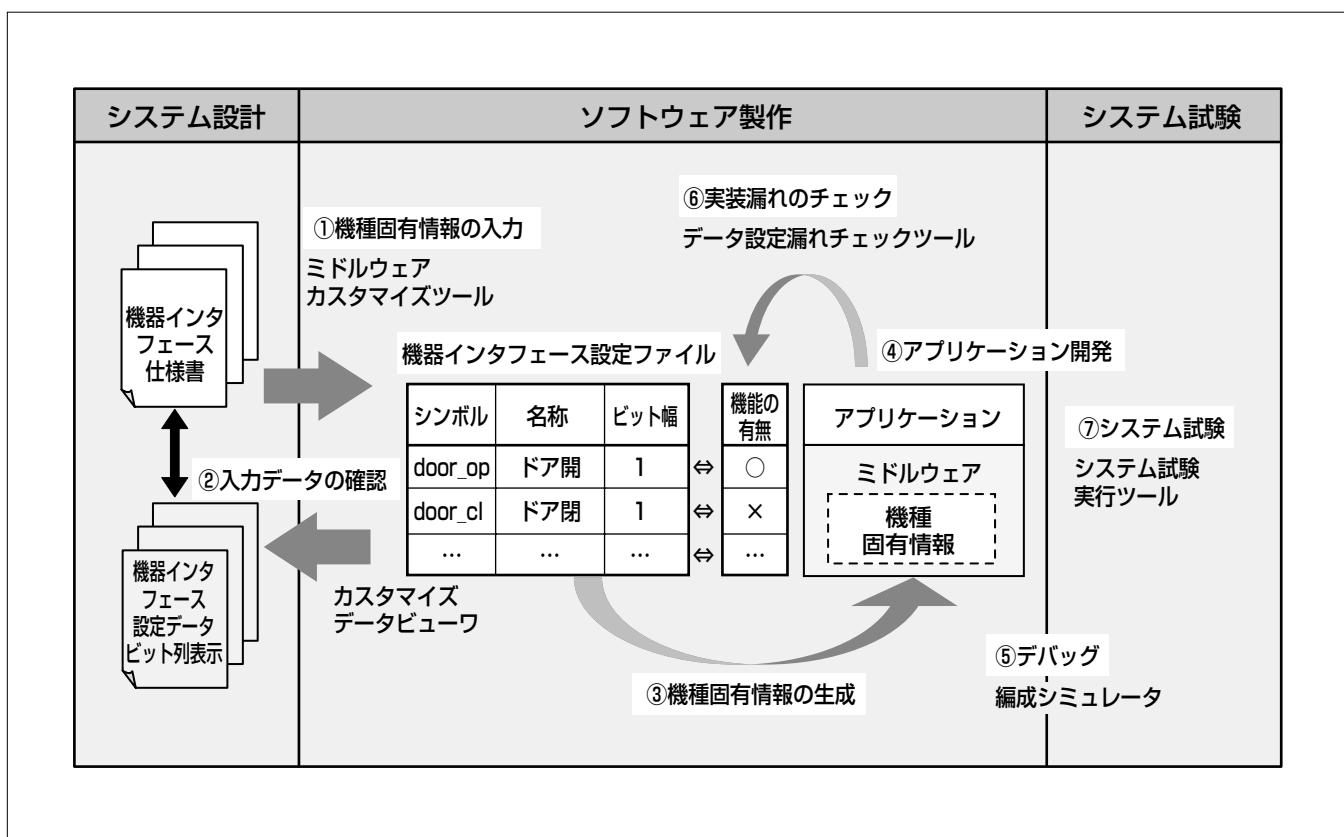
車両情報制御システムは、列車に搭載される機器全体を監視・制御する分散組み込みシステムであり、車両の仕様の多様性に追従した機能の柔軟性、拡張性を実現する上で、ソフトウェアが重要な役割を担っている。車両情報制御システムでは、機種ごとの仕様の差異が大きいため、従来は機種個別のソフトウェア開発が必要となっていた。

複数の機種にまたがるソフトウェア開発の効率化と品質向上のために、“PLATINA (Platform for Train Integrated management system Nucleus with Advanced technology)”を構築した。PLATINAは、ミドルウェア、開発ワークフロー、開発ツールなどを含むソフトウェア開発プラットフォームである。

PLATINAを構築することにより、以下を実現した。

- (1) 機種間で共通に利用する機能と変動する機能を分析した結果に基づき、ソフトウェア、開発ワークフローを再構築した。
- (2) 共通に利用する機能をミドルウェアとして実現し、ミドルウェア、及び、その上位のアプリケーションの再利用を可能とすることにより、新規ソフトウェアの製作量を削減した。
- (3) 開発ワークフローにおける各作業を支援するツール群を整備し、作業の効率化を図るとともに人為的ミス混入の可能性を低減した。

これらの総合的な取組みによって、開発効率の向上と信頼性の確保が達成できると考える。



## “PLATINA”におけるソフトウェア開発ワークフローと開発ツール

PLATINAは、機種間で共通に利用される機能をミドルウェアとして提供するとともに、ソフトウェア開発ワークフロー（図中の①～⑦）と、これらを支援する開発ツール群（ミドルウェアカスタマイズツール、カスタマイズデータビューア、編成シミュレータ、データ設定漏れチェックツール、システム試験実行ツール）を提供する。これによって、ソフトウェアの機種間での再利用が促進され、新規製作量が削減される。また、ツールによって人為的なミスの可能性が低減される。このため、ソフトウェア開発の効率化と信頼性の確保が達成される。

1. ま え が き

車両情報制御システムは、列車に搭載される機器全体を監視・制御する分散組み込みシステムであり、機能の柔軟性、拡張性を実現する上で、ソフトウェアが重要な役割を担っている。

これまでの車両情報制御システムの開発では、機種間で機能の類似性があるものの、列車の構成、接続する機器の多様性から、ソフトウェアを毎回新規作成する部分が大割割合を占めていた。近年、車両情報制御システムに接続される機器、要求される機能はますます拡大しており、ソフトウェアの製作量が増加している。このため、ソフトウェア開発効率と品質の向上が重要な課題となっている。

そこで三菱電機は、これらの課題に対応して、ミドルウェア、開発ワークフロー、開発ツールを含むソフトウェア開発プラットフォームPLATINAを構築した<sup>(1)(2)(3)</sup>。PLATINAを構築するに当たり、次の取組みを行った。

(1) 機種間で共通に利用する機能と、変動する機能を分析し、ソフトウェア、開発ワークフローを再構築する。共通に利用する機能をミドルウェア化し、ソフトウェアの機種間における再利用を促進することによって、新規作成量を減らす。

(2) ミドルウェアを核とした開発ツール群を構築する。  
本稿では、車両情報制御システムの構成について述べたあと、(1)を実現するための車両情報制御システムの分析、及び、この分析に基づき設計したミドルウェアの構成とその特徴について述べる。そして、(2)の開発ツール群についても述べる。

2. 車両情報制御システムの構成

車両情報制御システムは、演算ノード、表示器及びそれらを接続する伝送路で構成される(図1)。個々の演算ノードは、車両に搭載されるブレーキ、ドア、空調などの各機器と伝送、デジタル入出力などを用いてデータを授受する。

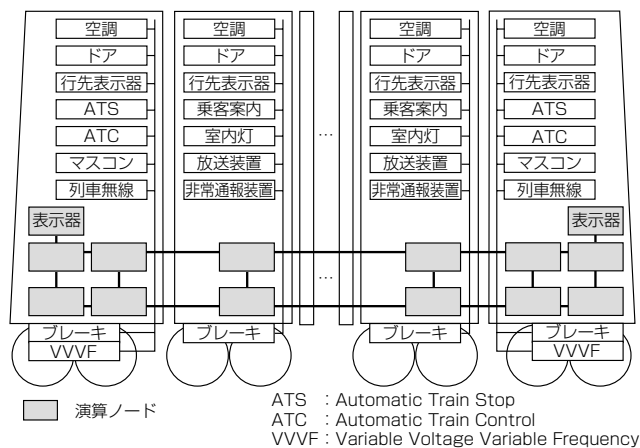


図1. 車両情報制御システムの構成例

列車内に複数設置された演算ノードは、伝送路で接続され、ネットワークを構成する。車両情報制御システムは、このネットワークを利用して、列車全体の監視・制御を実現する。

車両情報制御システムのネットワークは、信頼性を確保するために二重系で構成され、冗長性を確保している。演算ノードは対で配置され、演算機能が冗長化される。演算ノード、伝送路のいずれか一箇所の故障に対しては、故障箇所を迂回(うかい)する経路が存在するよう構成される。

3. 車両情報制御システムの変動性と共通性

ここでは、車両情報制御システムの変動性、共通性について述べる。

3.1 変動性

3.1.1 列車構成

車両情報制御システムの機種間の要求仕様、システム構成などに関する変動性について分析した結果、列車構成の変動性が、機種間でのソフトウェア再利用を妨げる要因の一つであることが分かった。列車構成とは、列車全体の編成、車両、機器の構成である。編成は、複数の車両の集合である。車両には、複数機器が搭載される。車両は、搭載される機器の種類、数に応じて決まる属性を持つ。

列車構成の機種間での変動は、演算ノード間の伝送パケットのフォーマット、機器と演算ノード間で授受する伝送パケットのフォーマットに影響を与える。これは、実時間性を実現するために設けたパケットの容量制限を満たすように、列車構成に応じてパケットのフォーマットを最適化する必要があるためである。また、列車構成が変化すると、演算ノードで実行されるソフトウェアが参照するデータの位置や、演算ノードの種類による処理実行の要否などが変化する。

3.1.2 演算ノード上のデバイス

機種によって演算ノードが入出力に用いるデバイスや、不揮発メモリデバイスが変更されることがある。この場合、デバイスドライバの変更が必要となる。

3.2 共通性

3.2.1 冗長構成と制御方法

共通性として、車両情報制御システムが監視制御する機器との伝送における冗長構成とその制御方法、演算ノードの冗長構成とその制御方法、編成の併合・分割時の処理方法など、インフラ部分に関する制御方法が挙げられる。

3.2.2 アプリケーション機能の共通化

変動性として挙げた列車構成を抽象的に表現することによって、アプリケーション機能を機種間で共通化できるものがある。例えば、パンタグラフは、機種によって、搭載位置、個数は異なる。しかしながら、個別のパンタグラフの上げ下げの制御方法や、状態の監視方法は機種間で共通化できる場合がある。また、その処理を実施する演算ノード

ドの指定も、“パンタグラフを搭載しているノード”という抽象的な指定方法をとることによって、共通化できる。

### 3.2.3 アプリケーション機能の枠組みの共通化

異なる機種間でも共通して利用されるアプリケーション機能が存在する。例えば、監視制御対象の機器との伝送データに応じて、あらかじめ決められた条件によって故障を検知し、それらを記録する故障検知機能、保守時に監視制御対象の機器に対して試験動作を指示し、結果を収集する機能などである。

これらの機能で、個別の条件や試験動作の内容は機種が異なれば変更される可能性があるが、それらを組み合わせることでアプリケーション機能を実現する枠組みは共通に利用可能である。

## 4. ミドルウェア適用によるソフトウェアの再利用

3章で述べた車両情報制御システムの変動性・共通性の分析に基づき、機種間の共通機能をミドルウェアとして構築した。ここでは、この構成と、ミドルウェア適用によるソフトウェアの再利用について述べる。

### 4.1 ソフトウェア構成

車両情報制御システムのソフトウェアは、ミドルウェアとアプリケーションで構成する。

#### (1) ミドルウェア

ミドルウェアは複数の機種で共通に利用される“共通部”と、機種ごとに変化する“機種固有情報”で構成される。共通部は、データ層、インタフェース処理層、ドライバ層から構成され、機器伝送、車両間伝送、CPU(Central Processing Unit)間伝送などの基本機能を持つ。また、ミドルウェアの管理するデータにアクセスするための共通のインタフェースとして、データアクセス関数がアプリケーションに提供される。機種固有情報は、列車構成情報や、演算ノードと機器間で共有されるデータのフォーマット情報などを含む。

#### (2) アプリケーション

車両情報制御システムのアプリケーションは、ミドルウェアが提供するデータアクセス関数を用いて、演算ノード間、及び演算ノードと機器間で授受するデータにアクセスして実現される。

### 4.2 ソフトウェアの再利用

ミドルウェア適用によって、複数機種を順次開発する際にソフトウェアの製作量が減る理由について、アプリケーションの機種間における再利用、及びミドルウェア化による再利用の観点から述べる。

#### 4.2.1 アプリケーションの機種間での再利用

従来、機種ごとに変動する機種固有情報はアプリケーションが個別に保持していた。このため、異なる機種を開発する際、機種固有情報の変更が必要であり、関連するアプ

リケーションすべての修正が必要であった。

一方、ミドルウェアは、上記機種固有情報を一元管理し、アクセス方法を標準化する(図2)。この仕組みを利用して実装されたアプリケーションは、機種固有情報を持たない。このため、異なる機種を開発する際の機種固有情報の変更が、アプリケーションに及ばないため、アプリケーションが再利用可能となる。

#### 4.2.2 ミドルウェア化による再利用

機器との伝送における冗長構成とその扱い方、演算ノードの冗長構成とその制御方法、編成の併合・分割時の処理方法は、機種間で共通に扱える機能である。ミドルウェアがこれらの機能を実現することによって、ミドルウェアそのものと同時に再利用される。

車両情報制御システムでは、監視制御対象を制御する系の切替え時や、編成状態が確定するまでの過渡的な状態であっても安定して動作することが求められる。これらの状況では、想定すべき状況が多岐にわたるため試験ケースが複雑になり、試験漏れの発生や、試験時間の増大が問題となる。

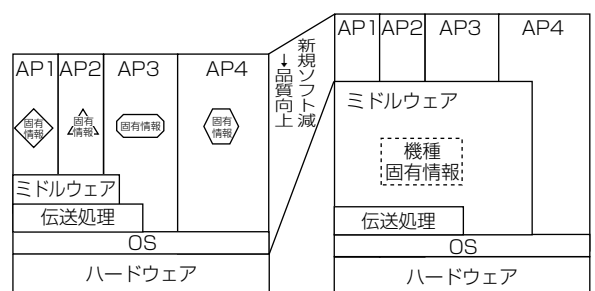
この機能をミドルウェアとして実現することによって、ミドルウェアに付随する試験ケースによって試験漏れの発生を抑制できる。また、これらの機能を新規実装する必要がなくなり、不具合混入の可能性が低減する。

## 5. ソフトウェア開発環境

車両情報制御システムの開発ワークフローは、システム設計、ソフトウェア製作、システム試験から構成される。ミドルウェア適用に合わせ、ソフトウェア製作段階に後述するツール群を開発し、適用できるようにした。これによって、開発の効率化と品質の確保が可能となる。

次に開発ワークフローに含まれる主な作業を示す。また、各作業で適用するツールもあわせて示す。

- ① 機種固有情報の入力(ミドルウェアカスタマイズツール)
- ② 入力データの確認(カスタマイズデータビューア)
- ③ 機種固有情報の生成(ミドルウェアカスタマイズツール)
- ④ アプリケーション開発



(a) 従来

(b) ミドルウェア適用時

図2. ミドルウェア適用によるアプリケーションソフトウェアの新規製作量削減

- ⑤デバッグ（編成シミュレータ）
- ⑥実装漏れのチェック（データ設定漏れチェックツール）
- ⑦システム試験（システム試験実行ツール）

次に、各ツールについて述べる。

5.1 ミドルウェアカスタマイズツール

ミドルウェアカスタマイズツールは、機器インタフェース仕様をデータ化し、機種固有情報の定義データを自動生成するツールである。

表形式による入力作業であり、ソフトウェアに詳しくない担当者でもデータ入力が可能である。また、入力データの一覧性を高め、入力間違いを防止する。

5.2 カスタマイズデータビューア

カスタマイズデータビューアは、ミドルウェアカスタマイズツールで入力したデータを、仕様書と同様のビット列表示で出力するツールである。ミドルウェアカスタマイズツールへの入力データと仕様書の内容を照合することによって、入力漏れやビットのずれなどの人為的ミスを低減できる。

5.3 編成シミュレータ

編成シミュレータは、汎用パソコン上で実機と同じアプリケーションを動作させ、デバッグを可能とするものである。

実機を用いてデバッグをする場合、処理タイミングが正確である一方、複数人での並行開発に限界があった。

アプリケーションは、複数の演算ノード上に搭載され、それらが連携して機能が実現されることが多い。このため、編成シミュレータは、編成に属する複数の演算ノードの動作を模擬する。

図3に編成シミュレータの構成を示す。車両情報制御システムのソフトウェアにミドルウェアを適用したので、演算ノードと機器間で共有されるデータへのアプリケーションによるアクセス方法が統一される。このため、シミュレータ用ミドルウェアを用意し、実機と同一のデータアクセス関数を提供することによって、汎用パソコン上で実機と同じアプリケーションを動作可能とした。

5.4 データ設定漏れチェックツール

アプリケーションは、ミドルウェア上に定義されたシンボルを用いて演算ノードと機器間で共有されるデータにアクセスする。データ設定漏れチェックツールは、シンボルを用いたデータ書き込み処理がソースコード上にあることを調査する。これによって、アプリケーションによるデータの設定漏れを検出し、人為的ミスを排除する。

5.5 システム試験実行ツール

車両情報システムが実現する機能の増大に伴い、試験項目数も増大しているため、試験操作時に混入する人為的ミスの削減と、試験実施の効率化の必要性が高まっている。

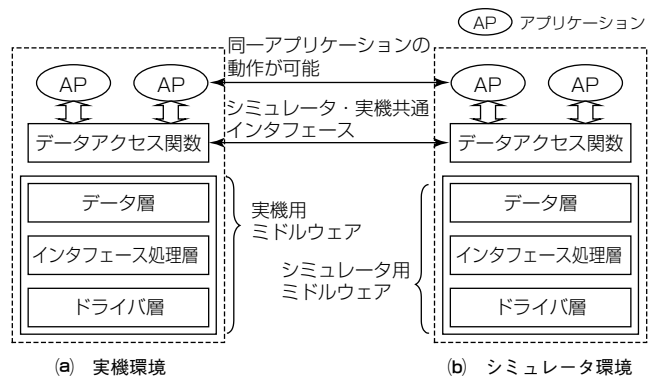


図3. 編成シミュレータの構成

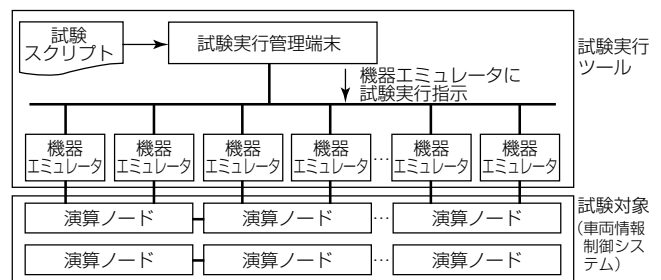


図4. システム試験実行ツールの構成

システム試験実行ツールは、与えられた試験スクリプトに従って、試験実行管理端末が機器エミュレータに対して動作を指示するツールである(図4)。

システム試験実行ツールは、車両情報制御システムが監視制御する機器の入出力を模擬する機器エミュレータと、それらの入出力動作を制御する試験実行管理端末から構成される。機器エミュレータに対する試験操作を、人手を介さずに実行できるため、試験操作における人為的ミスを排除するとともに、試験工数が削減される。

6. む す び

車両情報制御システムの機種間での共通性、変動性に基づき設計したミドルウェア、それを利用した開発ワークフロー、及びミドルウェアを核とした開発ツール群について述べた。

これらの総合的な取組みによって、ソフトウェアの生産性、品質の向上が達成できると考えられる。

参 考 文 献

- (1) 吉田 実, ほか: 列車情報管理装置のソフトウェアプロダクトライン, 三菱電機技報, 77, No.7, 479~482 (2003)
- (2) 小島泰三, ほか: 戦略的再利用に基づくソフトウェア開発, 三菱電機技報, 80, No.10, 647~650 (2006)
- (3) 辰巳尚吾: 列車統合管理装置のソフトウェア開発技術, システム制御情報学会セミナー (2005)