鉄道運行管理システム向け共通ソフトウェアアーキテクチャの開発

角本喜紀[†] 西島英児[†] 水津宏志[††]
先崎隆[††] 薦田憲久[†††]

LSI 技術の進歩にともない，監視制御系システムにおいても汎用製品の利用や効率的なソフトウェア開発が浸透しつつあり，鉄道運行管理システムにおいてもソフトウェア開発の効率向上がこれまで以上に強く要求されている．本論文では，鉄道運行管理システムにおいて，様々なシステム構成に共通に適用可能とするソフトウェアアーキテクチャとこれを実現するミドルソフトを提案する．提案ミドルソフトは，受信データのバッファリングと提示処理を 1 つの処理モジュールにて実行することにより排他制御のない高信頼な構造を実現する．また運行情報をシステム構成に依存せずつねに鉄道路線全体の情報として鉄道運行管理アプリケーションに提供する共通データインタフェース機能を持つ．

Development of Common Software Architecture Applicable to Various Types of System Structure in Train Traffic Control Systems

Yoshiki Kakumoto,[†] Eiji Nishijima,[†] Hiroshi Suizu,[††]
Takashi Massaki[††] and Norihisa Komoda[†††]

Software development considering software productivity and utilizing COTS (Commercial Off The Shelf) products has become popular in the field of control systems. Software development in train traffic control systems therefore needs to be much more effective than before. Software architecture and its middleware are proposed that will achieve the standard software necessary to be applied to various types of system structure. The proposed middleware has a reliable structure integrating a received data buffing function and data copying function into one program module. It also presents the operational data to applications as the entire train control area information regardless of system structures.

1. はじめに

鉄道運行管理システムは，運行指令や運行計画に従い信号機や転てつ器などの地上設備を制御して列車の進路を設定する監視制御システムで，これまでモノレール[1)]，地下鉄[2)]，大都市圏鉄道[3)] など様々な鉄道路線に導入されてきた．システムの一般的特徴は，数百ミリ秒から数秒程度の周期応答性が必要[2)]，運行乱れ時は運行計画のオンライン変更などで処理負荷が大きく増加[4)]，移動体監視のため監視情報に関して制御装置間で不整合が生じる場合があり一貫した情報を得るために整合化処理が必要[5)] などである．

この鉄道運行管理システムには制御対象となる鉄道路線の規模によって，中央集中制御型システムや駅分散制御型システムなどいくつかのシステム構成が存在する．ただし，それぞれのシステムは，対象となる鉄道路線向け専用のシステムでありソフトウェアもそれ専用に開発されてきた．このため，中央集中制御型のソフトウェアを駅分散制御型へ適用するなど，構成の異なるシステムへ適用することは非常に困難であった．

その一方で LSI 技術の進歩にともない，監視制御システムにおいても汎用製品の利用[6)] や効率的なソフトウェア開発[7)] が浸透しつつあり，鉄道運行管理システムでもソフトウェア開発の効率向上がこれまで以上に強く要求されている．よって本論文では，様々なシステム構成に対して共通に適用可能とする鉄道運行管理ソフトウェアの実現を目的とする．そのためには，次に述べる 2 つの課題を満足する必要がある．

第 1 に鉄道システムは，高い信頼性を確保するソフ

† 日立製作所システム開発研究所
Systems Development Laboratory, Hitachi, Ltd.
†† 日立製作所水戸交通システム本部
Mito Transportation Systems Product Division, Hitachi, Ltd.
††† 大阪大学大学院情報科学研究科
Graduate School of Information Science and Technology, Osaka University

トウェア構造を持つ必要がある．従来の鉄道運行管理システムでは，処理モジュール間の通信はメモリ上の共有データを介して行われ非同期にアクセスする場合，排他制御[8]が必要となる．しかし，アクセス順序などの状況次第ではデッドロック[9]を引き起こす可能性があり，排他制御のないソフトウェア構造を実現することが望まれる．排他制御を排する方式として，システム全体の起動管理をタイムトリガにより同期をとって管理するアーキテクチャ[10),11)]が提案されている．しかし，鉄道運行管理システムでは個々の装置がフリーラン実行することを前提としており，類似の起動管理を導入することは困難である．

第2の課題は，鉄道運行管理アプリケーションに対して，システム構成が異なってもつねに共通の運行情報を提供するデータインタフェースを実現することである．システム構成に依存しない位置透過な通信方式として，自律分散通信プロトコル[12)]や分散オブジェクト[13)]がある．しかし，これらは物理的なシステム構成を隠蔽できるが論理的なデータ提供機能を持たない．またビル管理[14)]やホームネットワーク[15)]では，機器単体ごとのデータインタフェースをデータオブジェクトとして標準化し提供している．しかし，鉄道運行管理システムでは，運行情報を論理的なテーブル形式で提供するデータインタフェースであり，さらに運行情報の整合化など移動体監視特有の処理が必要で，これらの技術をそのまま導入することはできない．

本論文では上記の課題に対し，複数のシステム構成に適用可能な装置共通のソフトウェアアーキテクチャと，これを実現するミドルソフトを提案する．ミドルソフトは，受信データのバッファリングとこれの提示機能を1つの処理モジュールで実行し，受信処理とアプリケーション処理との連接を排他制御のない構造で実現する．また運行情報の整合化機能を持ち，アプリケーションに対してシステム構成に依存しない共通のデータインタフェースとして鉄道路線全体の運行情報を提供する．

2. 鉄道運行管理システム

2.1 システム構成

鉄道運行管理システムは，軌道回路などの地上設備の状態を監視することにより列車の位置を確認し，運行計画や指令員の指示に従い信号機や転てつ器を制御して列車の進行進路を設定・確保し，あわせて列車の到着や出発の案内を行う監視制御システムである．

図1に鉄道運行管理システムの基本構成の1つである駅分散制御型を示す．ここでは，駅サブシステム側

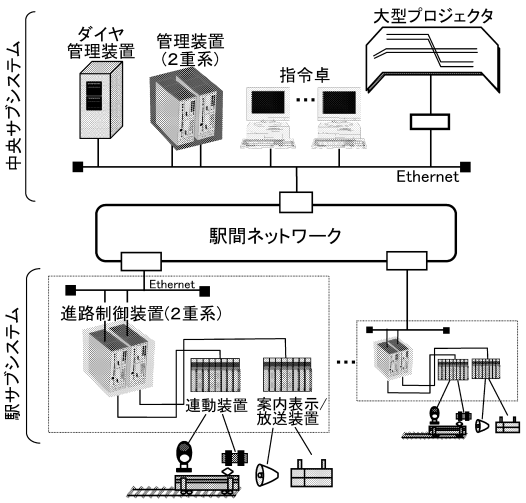

図1 駅分散制御型システム構成
Fig. 1 Distributed type system structure.

に列車の進路制御を行うための進路制御装置が存在し，各装置は決められた鉄道路線エリア内の制御を実行する．そのほか，信号機や転てつ器を管理する保安装置である連動装置，案内放送や案内表示用の装置が設置されている．中央サブシステム側には，指令卓，運行状況の監視や当日の運行計画を管理する管理装置などがある．駅分散制御型以外のシステム構成として，進路制御装置が中央サブシステムのみに1台ある中央集中制御型，中央サブシステムに複数台ある中央分散制御型システムがある．それぞれのシステム構成をどのような規模の鉄道路線に適用するかの目安は，中央集中型は小規模向け，中央分散は中規模向け，駅分散型は大規模向けといえる．なお汎用製品の利用が進んでおり，ネットワークは光IP通信やイーサネット，ハードはパソコンや汎用組み込み機器が利用されている．

2.2 進路制御機能

進路制御装置は，システム構成の違いにより配置場所や数が異なることを述べた．本節では進路制御装置の機能概要について述べる．進路制御装置における従来のソフトウェア構造を**図2**に示す．処理は大きく2つに分けることができ，(a) ネットワークから列車情報をイベント的に受信・登録する処理と，(b) 周期的に列車の追跡や進路の設定を行う処理である．処理 (a) は，列車情報受信処理が該当し，管理装置から運行計画情報や指令情報，他の進路制御装置から送信された列車追跡情報を受信・登録する．処理 (b) は，進路制御装置の主要な処理で，周期的に複数の処理モジュールを図2に示した番号の順に実行していく．該当の処理モジュールは，① 設備の情報を更新する設備情報

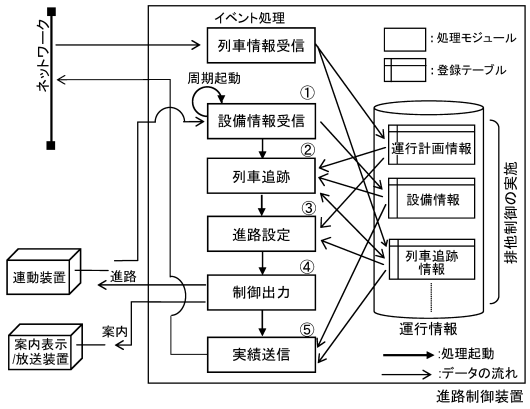

図 2 進路制御ソフトウェア構造
Fig. 2 Software structure in a route control machine.

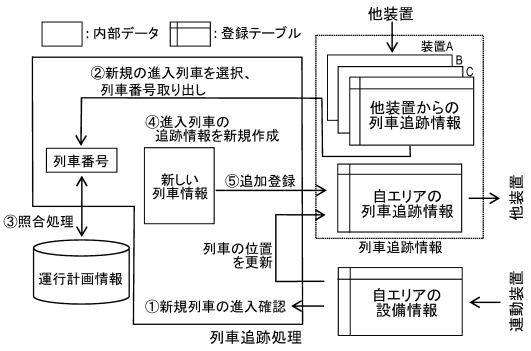

図 3 列車追跡情報生成
Fig. 3 Process of generating train tracking data.

受信処理，② 設備情報より列車の追跡を行う列車追跡処理，③ 運行計画情報に従い列車に対して設定すべき進路を決定する進路設定処理，④ 進路を確保するため信号機や転てつ器の制御情報を出力する制御出力処理，⑤ 設備や列車追跡の更新情報を送信する実績送信処理，より構成され，この一連の周期処理は数百ミリ秒から数秒の周期で実行される[2]．(a) のイベント的に実行される処理と，(b) の周期的に実行される処理は，それぞれ異なったタイミングで非同期に実行されるので，運行情報にアクセスする際，それぞれの処理モジュールは個別に排他制御を実行している．よって，提案アーキテクチャの第 1 の狙いは，排他制御のないソフトウェア構造をミドルソフトが提供することである．

さて，以上の処理モジュールのうち列車追跡処理は，集中制御型と分散制御型で機能に違いがあり，分散制御型の場合は進路制御装置間での連係機能を必要とする．**図 3** は分散制御型の処理フローを示す．処理フローは番号に対応して，① 更新された設備情報より自制御エリア内に新たな列車が進入したことを確認すると，② 他の進路制御装置からの列車追跡情報より新たに進入した列車情報を選択し，これの列車番号を取り出す．次に，③ 新たに進入した列車の列車番号と運行計画情報との照合処理を実施し，次に進入すべき列車であることを確認する．確認後，④ 該当列車の列車追跡情報を新規に作成し，⑤ 自制御エリア内に存在する列車の追跡情報として追加登録する．これにより該当の進入列車の列車追跡が開始される．自制御エリア内の列車追跡情報は，他の進路制御装置へ送信され，他の進路制御装置において同様に利用される．

このように分散制御型システムでは，複数の進路制御装置が，互いに列車の追跡情報を共有し列車の追跡を連係して実行する必要があり，これを列車追跡処理の機能として実現している．一方，集中制御型システムでは，進路制御装置が 1 台であり，列車追跡処理には進路制御装置間の連係機能が存在せず，設備情報より列車の位置を順次更新する処理が主な機能となる．同様に管理装置や指令装置でも分散制御型の場合は複数の進路制御装置から設備情報や列車追跡情報を受信するので，これらの装置でもシステム構成により処理機能が異なっている．

よって，提案アーキテクチャの第 2 の狙いは，システム構成が異なってもアプリケーションに対して共通のデータインタフェースを提供する機能をミドルソフトが提供し，集中型ベースのアプリケーションソフトを共通ソフトウェアとして複数のシステム構成に適用可能とすることである．

3. 鉄道運行管理共通ソフトアーキテクチャ

3.1 提案アーキテクチャの基本構成

図 4 に提案するソフトウェアアーキテクチャの基本構成を示す．共通ミドルソフトは，複数の受信処理，運行情報収集処理，運行情報編集処理，実績送信処理から構成され運行情報を管理する．このうち核となる処理モジュールは，運行情報収集処理と運行情報編集処理で，以下の特長的な方式を採用する．

(1) **運行情報受信とアプリケーションとの連接構造**
イベント的に他装置から列車追跡情報などを授受する機能とこれを周期的に提示する機能を，運行情報収集処理として 1 つの処理モジュールに統合し，さらには列車追跡処理など周期的に実行されるアプリケーション処理を，本モジュールの情報提示機能を起点に起動することにより，情報受信や参照において排他制御を排する構造とする．具体的には図 4 において，各受信処理がイベント的に受信した情報を運行情報収集処理が，そのつど，いったんバッファ登録する．一方の周

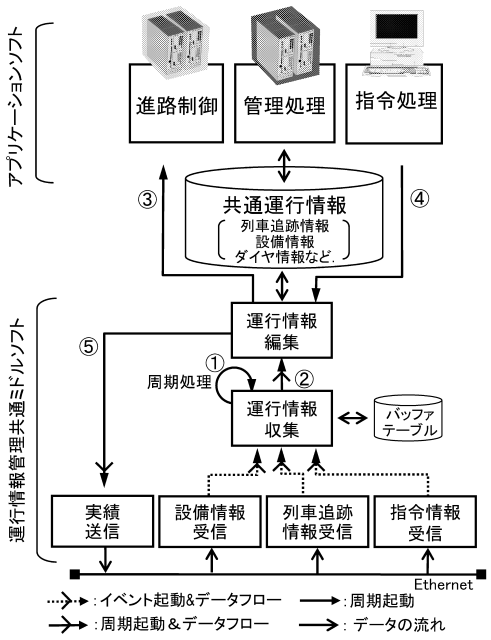

図 4 提案ソフトウェアアーキテクチャの基本構成
Fig. 4 Proposed software architecture.

期処理では，運行情報収集処理の周期起動開始（①）を起点として，バッファ登録された情報の取り出しと，図に示した順番（②〜⑤）での処理モジュールの実行を保証する．このとき，運行情報を情報種別ごとに管理しイベント処理機能と周期処理機能をモジュール統合する仕組みや，周期処理の処理シーケンスを保障する仕組みが必要となるが，詳細は 3.2 節で述べる．また運行乱れが起こる鉄道システムの特徴を考慮したうえで，周期処理の起動間隔を設定するなど，エンジニアリング的な分析も必要となるが，詳細は 4.1 節で述べる．

(2) **システム構成に依存しない共通インタフェース**
システム構成に依存しない位置透過な通信を実現するために機能コードによるマルチキャスト送信を特長とする自律分散通信プロトコル[12] を利用し，さらには運行情報の管理機能を運行情報編集処理としてミドルソフト化する．運行情報編集処理は，運行情報収集処理より提示された列車追跡情報や設備情報などの運行情報をシステム構成が異なっても鉄道運行管理アプリケーションに対してつねに共通の論理フォーマット，すなわち鉄道路線全体の情報（図 4 における共通運行情報）として提示する．これによりアプリケーションは，システム構成を意識せずに自身が管轄する鉄道路線エリアの情報のみを参照して監視や制御を実行すればよい．整合化機能などによる共通運行情報生成とデータインタフェースの詳細は 3.3 節で述べる．

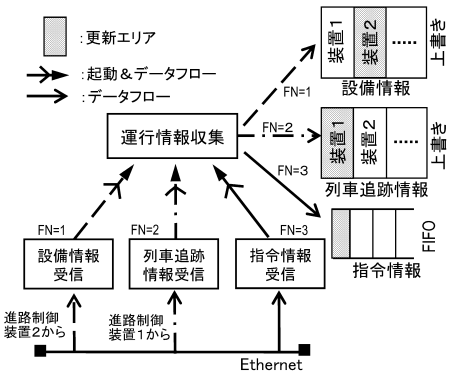

図 5 イベント受信によるバッファ登録機能
Fig. 5 Data buffering function in receiving event data.

3.2 運行情報収集処理

運行情報収集処理の特長は，各受信処理がイベント的に受信した情報を情報種別ごとにそのつどいったんバッファ登録する機能とこれを周期処理によって一括して提示する機能を，ファンクション番号（FN）を利用することにより 1 つの処理モジュールに統合したことにある．以下，それぞれの機能について具体的に説明する．

(1) **受信情報のバッファ登録機能**
受信処理では受信情報の種別に応じて受信処理モジュールを設けている．**図 5** に示すように各受信処理は，情報を受信するとただちに運行情報収集処理を該当の FN 指定で起動し，受信情報を運行情報収集処理に引き継ぐ．運行情報収集処理は各受信処理から起動されると，FN 番号，すなわち受信した情報の種別に従い受信情報を指定のバッファテーブルに登録する．列車追跡情報や設備情報の該当テーブルは，進路制御装置単位や連動装置単位のブロック構成となっている上書きバッファで，受信情報は情報を送信した装置の該当ブロックに登録される．指令情報などは，情報を順番に処理する必要があるので FIFO（First In First Out）型バッファである．

(2) **バッファ情報の提示機能**
バッファテーブルに登録されている情報を取り出すために，時刻管理の処理モジュールから FN 指定で周期的に起動されたときに実行される機能である．**図 6** に示すように，この機能ではバッファテーブルに登録された情報を運行情報編集処理が参照する参照用テーブルに一括して登録し運行情報編集処理を起動する．参照用テーブルの構成は対応するバッファテーブルと同じ構造である．つまり，上書きバッファの場合はデータのコピー処理を，FIFO 型バッファの場合はデータの遷移処理を実行する．

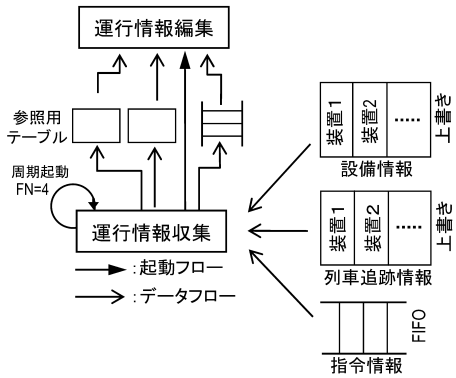

図 6 周期起動によるバッファデータ提示機能
Fig. 6 Data copy and shift function in cyclic triggered execution.

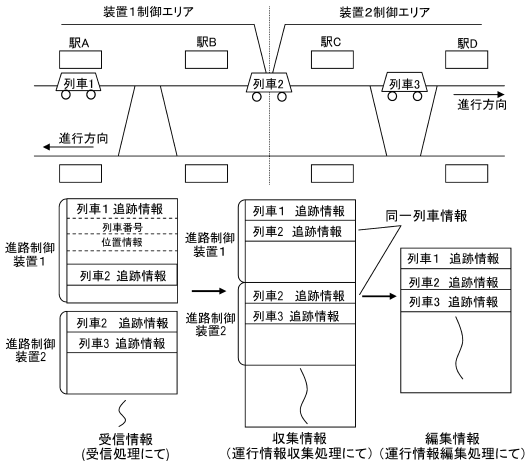

図 7 列車追跡情報の整合化処理
Fig. 7 Data integration process of train tracking data.

以上，FN 指定により起動タイミングが異なる複数の機能を 1 つの処理モジュールによって実行する構造としたことにより，これらの機能が並行して実行されることなく起動要求順に従って実行されるので，関連バッファや関連テーブルのアクセスにおいて排他制御の必要ない構造を実現することができる．また FN 指定に従った情報種別ごとの運行情報管理も可能となる．さらにはバッファリングにより運行情報編集処理の起動をスキップすることが可能であるので，万一 CPU 負荷が高まり図 4 に示した一連の処理を規定の起動時間間隔内で終了できない場合，運行情報収集処理は，運行情報編集処理の起動をスキップすることによりアプリケーション処理を含む処理シーケンスの連続性を保障することができる．

3.3 運行情報編集処理

運行情報編集処理は運行情報収集処理から起動され，運行情報収集処理が提示した列車追跡情報や設備情報などの整合化や統合化を行う．整合化は分散制御型で必要な機能で複数の進路制御装置から同一列車の列車追跡情報が送信された場合，列車の進行方向側の装置からの情報を優先して選択し列車情報を生成する機能である[5]．

図 7 に列車追跡情報の整合化機能を示す．図 7 の上部は路線状態を示し，路線の左右で制御を行う進路制御装置が異なり，列車 2 が制御境界エリア付近に存在する．このとき，図 7 下部に示すように両装置からの列車追跡情報に列車 2 が存在し，タイミングによっては両装置が示す列車の位置が異なる場合がある．この場合，整合化機能として運行保安を優先した論理によって列車の進行方向側の装置（図 7 では装置 2）からの情報を優先して選択する．この処理を全進路制御装置の情報に関して順次行うことにより鉄道路線全体

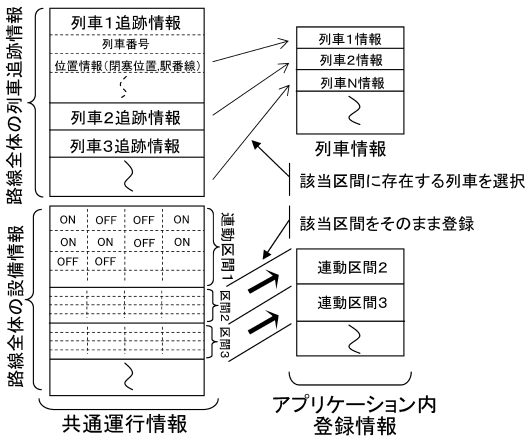

図 8 詳細データインタフェース
Fig. 8 Detail data interface.

の一貫した列車追跡情報を編集しアプリケーションに提示する．なお連動装置（図 1 参照）からの設備情報では，このような装置間での不整合は起こらず統合化のみ行う．この整合化機能や統合化機能を運行情報編集処理に実装することにより，運行情報編集処理はシステム構成に依存しない共通運行情報インタフェースを提供できる．また各装置が鉄道路線全体の情報を持つことにより特定の装置に依存せず装置独自で処理継続が可能となるため，システムとしてのフォールトトレランス性が向上する．

続いて，**図 8** に鉄道運行管理アプリケーションが共通運行情報にアクセスするデータインタフェースの詳細を示す．列車追跡情報は列車単位に登録されており，アプリケーションは，各列車追跡情報の位置に関するデータを参照して自身が担当する制御区間に列車が存

在するかを判定し，存在する場合はこれをアプリケーション内のプライベートエリアに登録する．路線全体が制御区間であれば，このような判断は必要なく全情報をそのまま登録する．設備情報の場合，連動装置が管理する設備機器の ON，OFF データが連動区間単位で登録されている．アプリケーションは，自身が担当する制御区間の情報を連動区間単位でプライベートエリアに登録する．以上の必要な共通運行情報を登録後，アプリケーションは処理を開始する．このデータ参照方式は各装置のアプリケーションソフトにおいて共通に実行可能である．

4. 提案アーキテクチャ評価

4.1 評価対象

システム稼動時における処理モジュールの起動検証と処理性能に関する評価を行い，提案アーキテクチャの有効性を定量的に検証する．評価対象は主要装置である進路制御装置で，CPU が 90 MHz の汎用マイコン，汎用の組み込みリアルタイム OS を使用した組み込み機器である．**図 9** は本システムの進路制御装置における関連処理モジュールの構成を示す．本装置では，連動装置から設備情報を受信する設備情報受信処理，他の進路制御装置から列車追跡情報を受信する列車追跡情報受信処理，そして指令情報など管理装置からの情報を受信する指令情報受信処理が，情報を受信したタイミングでファンクション番号（FN）指定により運行情報収集処理を起動し，これにより運行情報収集処理は受信情報のバッファリングを行う．一方，周期処理として番号（①〜⑦）に示した順序に従い処理モジュールが起動されていく．すなわち，運行情報収集処理や編集処理の起動により共通運行情報が制御アプリケーションに提示され，続いて列車追跡処理，進路設定処理，制御出力処理が順に起動されて進路設定情報が出力される．出力後，運行情報編集処理が起動され，これによりアプリケーション実行終了が通知される．これを受け運行情報編集処理は，実績送信処理を起動し，設備情報や列車追跡情報が外部の装置へ送信される．

続いて，周期処理の起動間隔設定に関するエンジニアリング的な分析について述べる．鉄道運行管理システムでは，運行乱れ時には運行計画のオンライン変更など指令情報が頻繁に発行され，これにより CPU 負荷が高まる．よって，通常運行時における CPU 負荷は 50%程度以下とされ[4]，これを満たすように周期処理の起動間隔を決定する必要がある．個々の処理の処理時間は，鉄道運行管理システムとしての特徴がある．

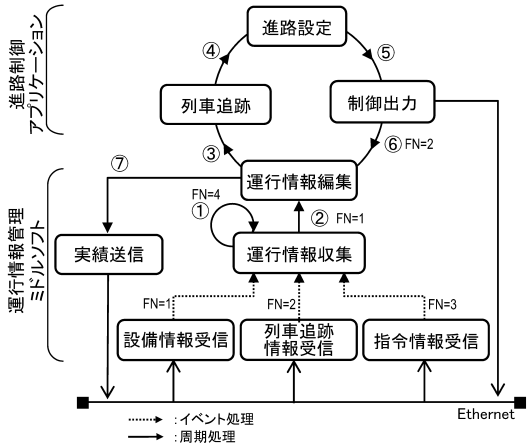

図 9 進路制御装置処理モジュール構造
Fig. 9 Task module structure in a route control machine.

たとえば，アプリケーション処理は監視エリア内の制御駅数，運行情報編集処理はシステム内の進路制御装置数にほぼ比例し，前記ハード仕様における処理時間は，アプリケーション処理では 1 駅あたり 10 ミリ秒から 15 ミリ秒，運行情報編集処理では進路制御装置 1 台あたり 5 ミリ秒から 10 ミリ秒を要する．これら数値よりシステム規模に応じた処理時間を積算して CPU 負荷が基準値を下回るよう起動間隔を概算し，評価システム試験や現地モニタラン試験を経て最終的に決定する．ただし，連動装置など接続している他装置とのインタフェースの関係上，CPU 処理能力に十分な余裕があっても起動間隔を短縮しない場合がある．

4.2 評価結果

まず，10 駅の制御を行う集中制御型システムにおける評価を述べる．集中制御型の場合，他の進路制御装置が存在しないので，図 9 の列車追跡情報受信処理は起動されることはない．**図 10** は起動ログデータ分析による関連処理モジュールの起動遷移を示す．周期処理では，最初に時刻管理処理モジュールが運行情報収集処理を起動している．この起動を起点に運行情報収集処理，運行情報編集処理，列車追跡処理，進路設定処理，制御出力処理が順番に実行されていることを確認できる．運行情報編集処理の実行中，他の処理モジュールに実行切替えが起きているが，これはより優先度の高い二重系相互監視処理に CPU の使用権が遷移したためである．相互監視処理が終了すると運行情報編集処理に戻って処理が継続されている．そして，制御出力処理が終了すると再び運行情報編集処理が実行されている．制御出力処理からの起動により，アプリケーションの終了を確認した運行情報編集処理は，実績送信処理を起動している．また適用システムでは

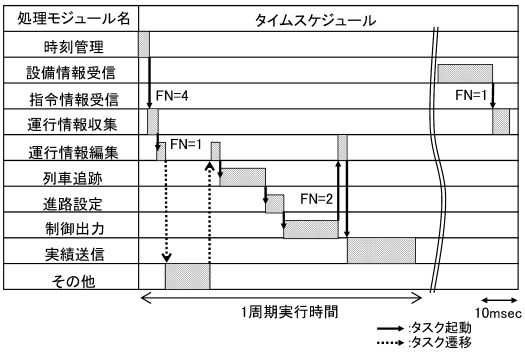

図 10 処理モジュール遷移図
Fig. 10 Time schedule of task modules.

1 秒ごとに連動装置から設備情報を受信する．このため時刻管理処理モジュールは，1 秒ごとに運行情報収集処理を起動するよう設定されている．図 10 の右側は設備情報受信時の処理起動の変遷を示している．設備情報受信処理は，データ受信後，運行情報収集処理を起動している．そのほか，イベント情報として管理装置からの指令情報があるが，処理遷移はこれと同様でデータ受信後，運行情報収集処理が起動される．

以上，ログデータ分析による処理モジュールの起動遷移の検証により，提案したソフトウェア構造が構築手順どおりに稼動していることを確認することができた．また周期 1 サイクルあたりの実行トータル時間が 100 ミリ秒程度であり，1 秒周期の起動間隔に対して処理性能上問題ないことを確認できた．

同様に分散制御型システムでは，進路制御装置 1 台あたりの制御駅数が 2 駅，合計 5 台あるシステムについて評価した．評価の結果，周期 1 サイクルあたりの実行トータル時間は 140 ミリ秒程度であった．列車追跡情報受信処理への実行切替え発生や運行情報編集処理で実行時間を要するが，進路制御装置 1 台が担当する制御エリアが狭いため，周期 1 サイクルあたりの実行時間は，集中制御型と同様 1 秒周期の起動間隔に対して処理性能上問題ないことを確認した．評価の結果から集中制御型や分散制御型とも処理性能面では周期起動間隔を短縮することが可能であることを確認したが，連動装置とのインタフェースの関係上 1 秒周期間隔としている．

5. おわりに

鉄道運行管理システムにおいて，複数のシステム構成に対して適用可能な共通のソフトウェアアーキテクチャとこれを実現する運行情報管理ミドルソフトを提案した．提案ミドルソフトは，システム構成にかかわらず共通運行情報インタフェースとしてつねに鉄道路線全体の運行情報をアプリケーションに提示する機能を持つ．またデッドロックを起こさない構造や高負荷時の周期処理スキップ機能を持ち高信頼なシステムを実現している．これにより，鉄道運行管理アプリケーションは，システム構成を意識せず自身が管轄する路線エリアに対して監視制御を実行すればよく，複数のシステム構成に対して共通の鉄道運行管理ソフトウェアの適用が可能となる．提案アーキテクチャは，いくつかの鉄道システムに適用されソフトウェア開発の効率向上や高信頼システム実現に貢献している．

ところで，組み込み機器分野のソフト生産性向上は大きな課題となっており，今後このようなソフトウェアの階層アーキテクチャ化はますます進展するものと予想される[16]．しかし，これまでの組み込み機器分野のソフトウェア開発は属人的要素が多分にあり，この分野の現状や課題が体系立てて紹介されなかったという背景がある[17]．本論文が組み込み機器分野の具体的な開発事例として関連研究者の参考となれば幸いである．

参 考 文 献

1) 幸田羊一，渡邊雅也，空 裕雅，前田哲夫：多摩都市モノレール輸送管理システム，第 36 回鉄道サイバネ・シンポジウム，pp.421–424 (1999).
2) 西村正明，冬木哲也，池上喜代一：京都市交通局烏丸線運行管理システム更新，第 37 回鉄道サイバネ・シンポジウム，pp.107–110 (2000).
3) 北原文夫，岩本孝雄，伊藤 聡，藤原和紀，藤原道雄：超高密度鉄道の列車群を自律分散制御する東京圏輸送管理システムの開発，電学論 D, Vol.118, No.4, pp.534–541 (1998).
4) 傳雄一郎，関灘俊雄，久松俊彦，冨田浩史，大隅英貴，長井 聡：大規模運行システムのアシュアランス技術，第 37 回鉄道サイバネ・シンポジウム，pp.115–118 (2000).
5) 角本喜紀, 西島英児, 石井賢一, 佐藤博明, 早乙女弘：鉄道運行管理システムにおける複数システム構成を実現する統一ソフトウェアアーキテクチャの開発，平成 12 年電気学会産業応用部門大会，pp.999–1004 (2000).
6) 鮫嶋茂稔，河野克己：オープン自律分散 FA システム，システム/制御/情報，Vol.43, No.1, pp.42–47 (1999).
7) 鵜林尚靖：組み込みソフトウェアの設計モデリング技術，情報処理，Vol.45, No.7, pp.682–689 (2004).
8) Zhao, W., Ramamritham, K. and Stankovic, J.A.: Preemptive Scheduling Under Time and Resource Constraints, *IEEE Trans. Comput.*, Vol.C36, No.8, pp.949–960 (1987).

9) Kopets, H.: *Real-Time Systems, Design Principles for Disributed Embedded Applications*, Kluwer Academic Publishers (1997).
10) Kopets, H., Kruger, A., Millinger, D. and Schedl, A.: A Synchronization Strategy for a Time-Triggered Multicluster Real-Time System, *Proc. 14th Symp. on Reliable Distributed Systems*, pp.154–161 (1995).
11) Berwanger, J., Ebner, C., et al.: FlexRay—The Communication System for Advanced Automotive Control Systems, *2001 SAE World Congress*, SAE Press paper 2001-01-0676 (2001).
12) Kawano, S., Kawano, K. and Wataya, H.: An Autonomous Decentralized System Architecture for Distributed Information and Control Systems, *Proc. COMPSAC '98*, pp.556–562 (1998).
13) 小野沢博文：分散オブジェクト指向技術CORBA，ソフトリサーチセンター (1996).
14) 中村政治：ANSI/ASHRAE BACnet プロトコルの概要と特徴，平成12年電気学会産業応用部門大会，pp.587–592 (2000).
15) ECHONET 規格書. http://www.echonet.gr.jp/
16) 菅沼賢治，村山浩之：自動車組み込みソフトウェアにおける開発戦略，情報処理，Vol.44, No.4, pp.363–368 (2003).
17) 高田広章：組込みシステム開発の要素技術と標準化，情報処理，Vol.46, No.4, pp.417–422 (2005).

(平成17年 2 月25日受付)
(平成17年11月 1 日採録)

角本　喜紀（正会員）

1964年生．平成元年京都大学大学院工学研究科応用システム科学専攻修士課程修了．同年（株）日立製作所入社．交通システム，防衛システムの研究開発に従事．現在，システム開発研究所に勤務．電気学会，プロジェクトマネージメント学会会員．

西島　英児（正会員）

1967年生．1986年徳島県立徳島東工業高等学校情報技術科卒業．同年（株）日立製作所入社．1990年日立京浜工業専門学院情報工学研究科卒業．鉄道トータルシステムの研究開発に従事．現在，システム開発研究所第一部勤務．電気学会会員．

水津　宏志

1973年生．1998年東京大学大学院工学系研究科修士課程（電子情報工学）修了．同年（株）日立製作所入社．主として運行管理システムの設計に従事．現在，水戸交通システム本部信号システム設計部勤務．

先崎　　隆

1955年生．1982年東京理科大学理学部（数学科）卒業．同年（株）日立製作所入社．主として運行管理システムの設計・とりまとめ業務に従事．現在，水戸交通システム本部信号システム設計部勤務．

薦田　憲久

1950年生．1974年3月大阪大学大学院工学研究科電気工学専攻修士課程修了．同年（株）日立製作所入社．システム開発研究所勤務．1991年4月大阪大学工学部情報システム工学科助教授，1992年8月同大学教授．2002年4月より，同大学大学院情報科学研究科マルチメディア工学専攻教授．工学博士．情報システムの研究に従事．電気学会2000年度進歩賞等を受賞．IEEE等の会員．