

列車統合管理システム向けGUIソフトウェア基盤 Quintetの開発

伊丹事業所 技術第1部 車両情報システム第3課
平野 耕一

三菱電機(株) 先端技術総合研究所 ソリューション技術部 プラント計装システム技術G
大崎 雅代

1. まえがき

列車統合管理システム(Train Control and Monitoring System) (以下、TCMS)は、列車の状態を監視・制御するシステムである。列車の運転台には表示装置(Display Unit) (以下、DU)が搭載され、乗務員はDUから列車の状態を監視・制御する。

TCMSは列車の中核部としての重要性を増しており、従来では分散型制御を適用していたが、ネットワーク技術の進展もあり、次世代車両制御システムに向け、さらなる機能向上、信頼性向上を図るために集中型制御が適用されるようになった。

この集中型制御のTCMSにDUを適応させるため、三菱電機(株)先端技術総合研究所と共同でTCMS向けGUI(Graphical User Interface)ソフトウェア基盤であるQuintet(Qt based ui and web integrated framework)を開発した。

本稿では、Quintetの概要について紹介する。

2. 開発の背景とコンセプト

2.1 分散型制御TCMS

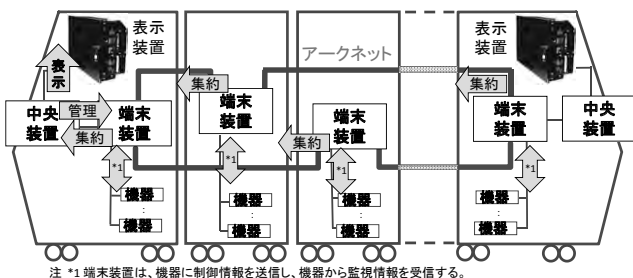


図1. 分散型制御方式のシステム構成図

分散型制御TCMSは、図1のように車両毎に制御を行うシステムであった。先頭車両の中央装置と車両毎に各機器を制御する端末装置から構成される。

伝送方式はアーキネット^(注1)(10Mbps)であり、ネットワークの通信帯域は狭い。そのため、図1のように端末装置が車両毎に制御を行い、列車としての状態を監視・制御するために、必要なデータだけを抽出して列車内ネットワークに流す方式を取っている。

この方式では、データ追加など伝送するデータ構造を変更した場合、データの抽出のため中央装置とDUの両方のソフトウェア変更が必要となる課題があった。

2.2 集中型制御TCMS

集中型制御のTCMSは、通信帯域が広いEthernet^(注2)(100BASE-TX)を採用し、更なる機能向上のために、分散型制御のTCMSから、図2のようなシステム構成に見直した。コンセプトは次のとおりである。

①集中型制御

中央装置に制御を集約することで車両単位制御から列車全体制御を実現した。これにより、端末装置はデータ転送のみ行う伝送装置に置き換えることができる。

②IEC61375^(注3):Train Communication Networkへの準拠

国際標準規格に準拠し、列車内の他システムとの相互接続性の担保を行う。

③信頼性の向上

TCMSに搭載される基板の高密度化、単機能化が行われた。車両毎の部品点数を削減することにより故障率を下げ、信頼性の向上が期待できる。

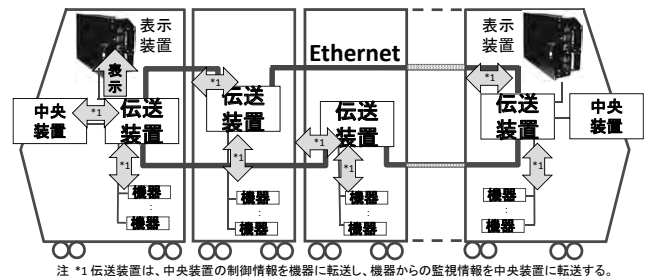


図2. 集中型制御方式のシステム構成図

2.3 TCMSにおけるDUの課題

このコンセプトで開発を進めるうえで、集中型制御TCMSに搭載するDUには下記の課題解決が必要であった。

(注1) アーキネットはネットワークの規格の1つである。

(注2) ネットワークの規格の1つである。

(注3) IEC61375は鉄道の車両制御システム規格である。

(1)分散型から集中型に変更することによる課題

①データの大容量化・通信速度の高速化への対応

分散型制御TCMSのDUは、データに変更が無い場合でも周期的に通信を行っている。データの大容量化・通信速度の高速化に対応するためには必要な時に必要な量のデータを取得することが必要になる。

②画面仕様とデータ仕様の独立性向上

分散型制御TCMSではDUは、中央装置が集約するデータのみを参照しており、機器との伝送データ仕様が変更されても集約データに変更無ければアプリケーションへの影響は無い。

しかし、集中型制御TCMSではDUは、全ての装置と伝送を行うため、伝送データ仕様変更の影響を大きく受ける。そのため、画面仕様とデータ仕様の独立性を向上させ、データ仕様変更の影響を小さくすることが必要になる。

(2)分散型制御TCMSから継続する課題

①多様化する併結パターンへの対応

分散型制御TCMSでは、システムの動作をシンプルにするために、分割・併結パターンの組合せを限定していた。しかし、車両の効率的な運用を実現するため、編成の分割・併結パターンは多様化する傾向にある。併結パターンの組合せ数が多い場合でも容易に対応出来ることが必要になる。

②複数のプラットフォームへの対応

DUのアプリケーションは、組込機器に搭載する場合や、乗務員の教育のためPC上に搭載する場合があります、案件ごとにWindowsやLinuxなど様々な環境下で動作することが必要になる。

これらの課題を解決し、システムに適したソフトウェア構成に見直すため、集中型制御のTCMS向けGUIソフトウェア基盤であるQuintetの開発に至った。

3. Quintetの概要

分散型制御におけるDUの課題解決のため、Quintetは次の方針で開発を進めた。

- (1)画面仕様とデータ仕様の独立性が強いアーキテクチャを構築する
- (2)使用するデータだけを抽出することでデータの大容量化に対応する
- (3)組合せ数の多い併結パターンに対応する
- (4)複数のプラットフォームに対応する

上記(1)～(4)の実現のため、Quintetの全体アーキテクチャを図3のように設計した。図3の各動作(①～⑩)を表1に示す。詳細は3.1節以降に示す。

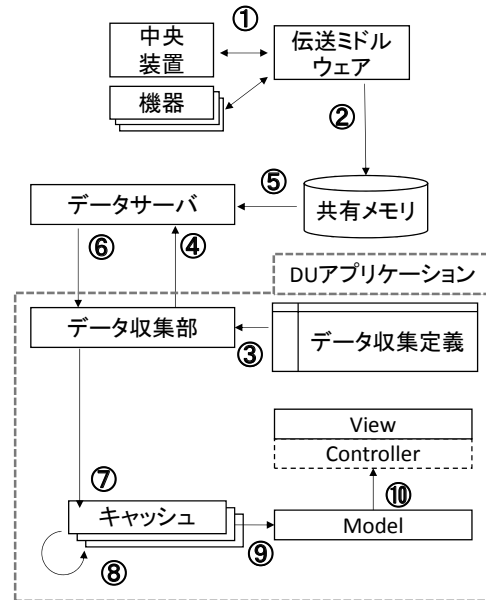


図3. Quintetの全体アーキテクチャ

表1. Quintetの動作

No	動作
①	中央装置と機器が発信するデータを伝送ミドルウェアが受信する。
②	伝送ミドルウェアが受信したデータを共有メモリに格納する。
③	データ収集定義から画面が必要としているデータを参照する。
④	データサーバにデータを要求する。(3.2節を参照)
⑤	共有メモリから要求されたデータを検索する。
⑥	データ収集部に要求したデータを配信する。
⑦	受信したデータをキャッシュに保存する。
⑧	画面表示に適した形にデータを加工する。
⑨	画面に表示するデータをModelに格納する。
⑩	画面にModelの内容を表示する(3.1節を参照)

3.1 画面仕様とデータ仕様の独立性向上

分散型制御TCMSのDUアプリケーションは、画面仕様とデータ仕様の結合度が高く、データ仕様の変更は画面表示にも影響を与える。そのため、Quintetは、データ収集部と表示処理を分離する設計とし、画面仕様が決まった段階で画面の描画部品(後述のView部品とModel部品)の製作を可能とした。

GUIの各機能の役割を明確化するデザインパターンの1つにMVC(Model View Controller)パターンがある。QuintetはMVCパターンを派生させViewとControllerを統合した、図4に示すModel/Viewアーキテクチャを採用した。テーブル構造でデータを保持するModelとそれを元に描画処理を行うViewに機能分割し、図3のように、データ

収集部をModelから分離することで、画面仕様とデータ仕様の独立性向上を実現した。

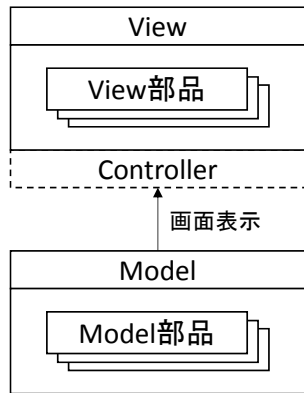


図4. Model/Viewアーキテクチャ

Viewは複数のView部品群、Modelは複数のModel部品群から構成される。View部品は、ボタンや表・図形など画面上に表示される部品を示す。Controllerは、画面表示への入力処理であり、このアーキテクチャではViewに統合している。Model部品は、行が状態を表し、列が車両を表す、図5のような仮想テーブルを持つ。

View部品は、Model部品に設定された内容に基づき、図5に示すように画面表示を行う。Model部品が更新されるたびにView部品にその結果が反映される。(表1 No⑩)

3.2 データ大容量化への対応

分散型制御TCMSのDUアプリケーションは、データに変更が無い場合でも周期的に通信を行っており、通信帯域を効率的に使用していなかった。Quintetはデータ大容量化に対応するため、イベント駆動で通信を行う仕組みを導入した。

イベント駆動型プログラムにおけるデザインパターンの1つに出版/購読型モデルがある。出版/購読型モデルは送信者(出版側)が受信者(購読側)を想定せずにメッセージを送るモデルである。出版側のシステムと購読側のシステムは、それぞれ相手の状態によらず動作可能である。

Quintetはアプリケーションと伝送ミドルウェアの間に出版/購読型モデルのデータサーバを設け、必要なデータのみ

を必要な時に参照する図6の仕組みを導入し、データ量増加に対応した。

分散型制御TCMSでは、DUは中央装置から受けた集約データに対して応答を返していた。集中型制御TCMSでは、DUは主体的にデータを要求し必要なデータを受ける。

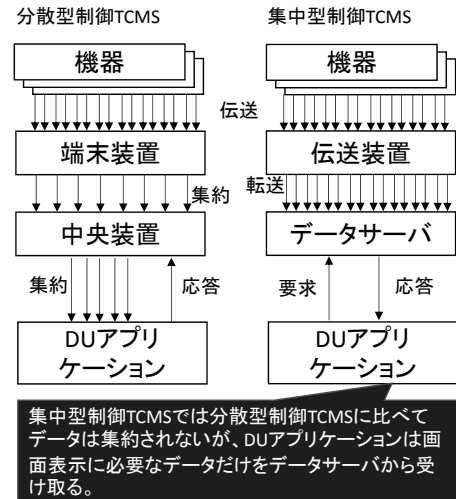


図6. データ量増加への対応

データサーバへの要求は、データ収集部がデータ収集定義の内容に基づいて行う(表1 No③)。データ収集定義には、パケット名称及びデータ名称を割り付ける。名称を定義するため、データの位置やサイズなどデータ構造には依存しない。そのため、受信するデータ量が増加してもアプリケーションは使用するデータのみを扱うことができる。

DUアプリケーションは、データ収集定義に記載されたデータをもとに、収集対象のデータを一意に特定可能なID(数値)に変換し、そのIDをデータサーバに要求する(表1 No④)。データサーバは、共有メモリ上からDUが要求するIDのデータを取得し(表1 No⑤)、DUアプリケーションに配信する(表1 No⑥)。データサーバへの要求は周期配信・イベント配信・周期+イベント配信のいずれかを選択可能とした。



図5. View部品とModel部品の関係

Quintetでは、出版/購読型モデルのデータサーバ導入によりイベント駆動でデータ更新及び画面描画を行えるようにした。具体的には、Model部品のデータが更新された時にView部品を更新している。これにより、DUアプリケーションは、画面表示に必要な情報を変化があった時のみデータサーバから取得可能とした。また、頻繁に変化するデータの場合は周期的に配信した方が通信帯域を効率的に使用できる場合もある。そのため、分散型制御のDUが行っていた機能と同様に周期的にデータを取得することも可能とした。

効率的なデータ取得を行うことで、データ取得までの時間を短縮し、分散型制御TCMSに比べて画面描画を高速に行えるようになった。

3.3 多様な併結パターンへの対応

列車は、運用中の任意のタイミングで併結・分割を行うため、車両数の動的な変化に対応することは客先の大きなニーズの1つである。システムにより編成認識方法は異なるため、Quintetは併結パターンの組合せ数が多い場合にも対応可能とした。

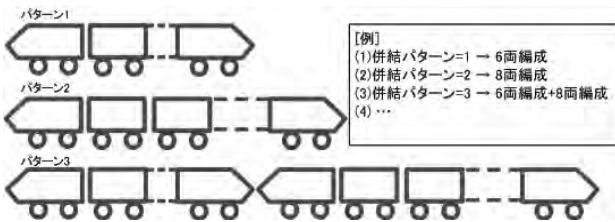


図7. 分散型制御TCMSにおける編成認識

分散型制御TCMSにおいてDUは、図7のように併結パターンをあらかじめ全て定義していた。DUは、列車構成情報を中央装置からしか得られず、中央装置からはデータを集約するために併結パターンの番号だけを得ている。そのため、組合せに無いパターンを追加するには中央装置とDUの双方に改修が必要になる。

集中型制御TCMSにおいてDUは、列車構成情報を中央装置と同様に取得可能である。列車構成情報は、TTDP (Train Topology Discovery Protocol) による編成認識処理により決定される。TTDPは、隣接する車両が存在するか車両のトポロジーを調べることで併結状態を認識するプロトコルであり、決定した列車構成情報はTCMSネットワーク内の各装置に通知される。DUアプリケーションは、この列車構成情報から車両数及び車両の並びを認識する。

Quintetでは、View部品とModel部品は仮想テーブルの1列を1車両として取り扱っているため、編成数の増減は図8のように仮想テーブルの列を増減させることで対応可能とした。

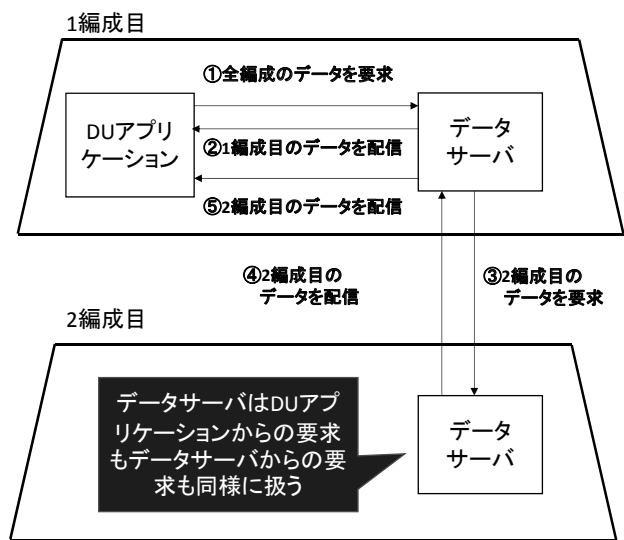


図9. データサーバの多段階購読

また、Quintetでは、データサーバへの問合せは現在の車両数分だけ行う。例えば6両編成の場合、1車両目から6車両目までのデータを収集対象とする。

併結構成の車両の場合、データサーバは編成ごとに配置され、図9のようにサーバ間で購読/配信を中継する多段階購読機能によりデータを収集する。

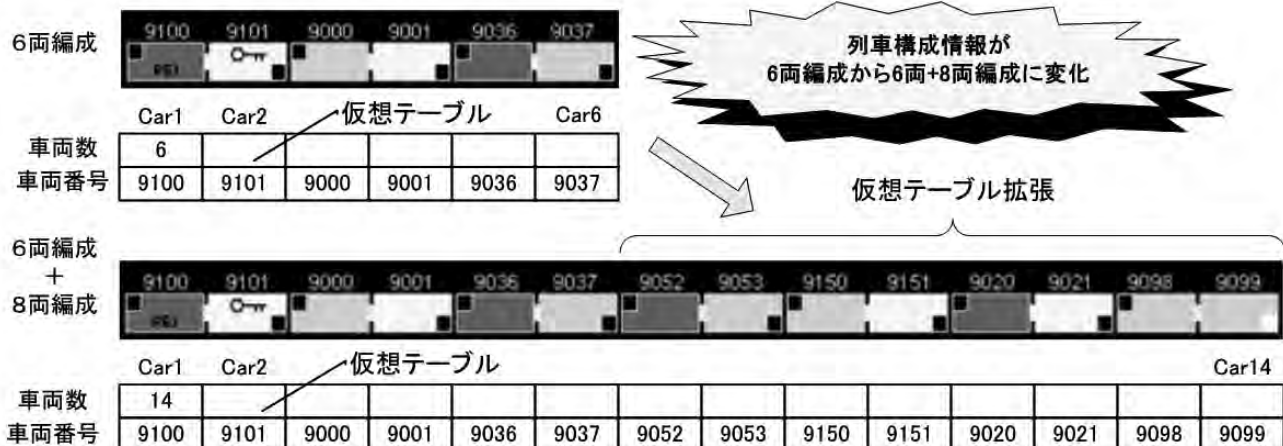


図8. 集中型制御TCMSにおける編成認識

DUアプリケーションは、併結構成の場合も単編成と同様に、自編成のデータサーバに対してデータの購読要求を行う。データサーバは要求されたデータが他編成にある場合、他編成のデータサーバからデータを取得してDUアプリケーションに返す。この時、データサーバはアプリケーションからの要求も他編成のデータサーバからの要求も同等に扱うため、DUアプリケーションはデータがどの編成のデータサーバにあるかを意識する必要はない。

例えば、図8のように6両編成+8両編成の場合、DUアプリケーションは自編成のデータサーバには14両分のデータを要求する。自編成内で集められないデータは他編成に中継し、自編成のデータサーバを介して、14両分すべてのデータを収集可能である。

3.4 マルチプラットフォーム化

Quintetの開発にはQtを採用した。Qtはそれぞれのプラットフォーム向けにビルドすることにより、同じソースコードで同じ機能を動作させるという特徴がある。

Quintetは、Qtを使用することによりマルチプラットフォームアプリケーションの開発を可能にした。

このため、組込機器に搭載するLinux上で動作するソフトウェアを、乗務員の教育用のソフトウェアとしてWindows上で動作させることも可能である。

4. 開発の成果

Quintetは、2016年から開発を行い、2017年9月時点で海外5案件に適用され、次のような成果があった。

(1) 画面表示性能の向上

Quintetは、データサーバの導入により大容量化したデータから必要なデータのみを抽出し、イベント駆動型の処理とした。分散型制御システムに比べてデータを取得するまでの応答時間が1/10に改善され、画面描画が高速化された。

(2) 伝送フォーマットへの依存度の低減

データ収集部と表示処理を分離し、MVCパターンの採用により表示処理をModel/Viewの構造に見直したため、DUアプリケーションは伝送フォーマットへの依存度を低減できた。

伝送フォーマットが未決定の段階でも表示処理は単独で動作確認が可能になった。

(3) 組合せ数の多い併結パターンに対応

Quintetでは、Model/Viewアーキテクチャとデータサーバを導入し、列車構成情報の取得方式を変更したこ

とで、列車運用中の車両数変化にも対応可能となった。そのため、組合せ数の多い併結パターンの場合でもアプリケーションは容易に対応可能になった。

(4) マルチプラットフォーム化の実現

Quintetは、Qtを使用することによりマルチプラットフォームアプリケーションの開発が可能になり、案件ごとに要求されるプラットフォームが変更されても対応可能になった。

5. むすび

本稿で紹介したGUIソフトウェア基盤のQuintetを適用することで、集中型制御システム構成に適した開発を進めることができるようになった。

今後も新規案件へのQuintet適用が見込まれており、競争力強化のため、更なる改善策を進めていく。

最後に、本開発にあたり技術的な助言をはじめとして、さまざまな面でサポートいただいた関係者の方々に深く感謝申し上げます。

執筆者紹介



平野 耕一 ヒラノ コウイチ
2007年入社。主に列車モニタリング車
上システムのソフトウェア開発に従事。
現在、伊丹事業所技術第1部車両情報シ
ステム第3課。



大崎 雅代 オオサキ マサヨ
1991年三菱電機入社。大規模ソフト
ウェアの開発効率化の研究開発に従
事。現在、先端技術総合研究所ソリュー
ション技術部プラント計装システム技術
グループ。