

空調制御ソフトウェアの標準化開発

長崎事業所 技術第2部 車冷システム課
津田 憲宏

1. まえがき

鉄道車両用空調装置(以下、空調装置)は三菱電機(株)長崎製作所(以下、長電)の主要製品であり、当所はその空調装置の空調制御ソフトウェアを開発している。

空調装置は鉄道会社や車両タイプごとに製品要件が異なるため、要件に合わせて、機器や機器構成が異なる製品を開発してきた。これに伴い、空調制御ソフトウェアも製品に合わせて開発を行ってきた。ソフトウェア開発を、低コスト、短納期で実現するため、開発する機能要件に類似する過去のソフトウェアを流用する派生開発スタイルを長年繰り返してきたので、派生したソフトウェアが増加した。このような開発を繰り返した結果、ソフトウェアはモジュール間の関係やインターフェースが複雑になり、改造時の影響範囲の判断が難しくなっていた。

今回、空調装置の機器構成のパターン化や複雑化したソフトウェアの見直しにより、更なる低コスト、工期短縮を図るため、空調制御ソフトウェアの標準化開発を行った。目標をソフトウェア開発全体で30%の工数削減と設定し、ソフトウェア開発プロセスには、組込みシステムで注目されている「モデルベース開発」を適用した。

本稿では、標準化開発における、ソフトウェア部品化の設計方針、モデル化の範囲と展開方法及びMATLAB/Simulink(以下、MATLAB)の効果を中心に紹介し、最後に標準化開発における改善効果を述べる。

2. 空調装置の概要

2.1 空調装置

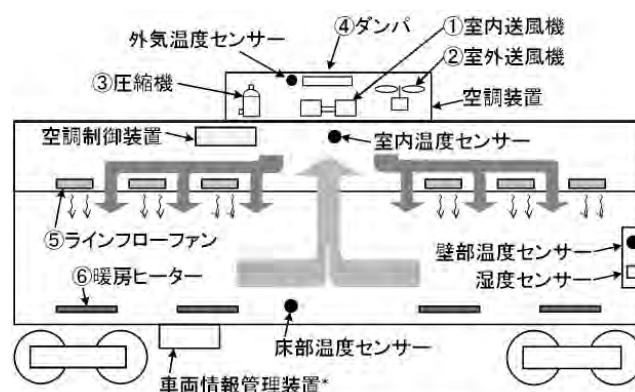
一般的な電車を例に、車両内の空調装置の機器構成を図1に示す。

空調装置は、室内送風機(図1①)、室外送風機(図1②)、圧縮機(図1③)、ダンパ(図1④)により構成される。その他システム構成として、車両に設置されるラインフローファン(図1⑤)、暖房ヒーター(図1⑥)がある。

空調装置では、冷凍サイクルの原理により、冷房を行っている。圧縮機により圧縮した冷媒ガスを循環させ、室外送風機により車内の熱を車外に排出し、室内送風機により車内に冷風を吹き出す。

暖房ヒーターは、車両の床や座席下に設置される。ラインフローファンは、天井内に設置される線状の扇風機で、

空気のかくはんや涼感を与える役目がある。



注*:空調装置を含む各種車載機器を管理する装置

図1. 車両内の空調装置の機器構成

空調装置は、鉄道会社や通勤電車、豪華列車などの車両タイプで、構成する機器の台数が異なる。室内送風機、室外送風機、圧縮機の台数を変え、車両タイプに応じた制御を行うことで、最適な空調制御を実現している。

2.2 空調制御ソフトウェア

空調制御ソフトウェアの機能構成を、図2に示す。

空調制御ソフトウェアは、①外部インターフェース、②温度制御、③機器制御、④保護機能、⑤故障記録により構成される。以下に空調制御の機能を説明する。

- ①外部インターフェースで車両情報管理装置からの空調制御情報を受信し、冷房か暖房か(空調運転モード)を決定する。
- ②温度制御で車内温度と目標温度の温度差により冷暖房の強さ(運転能力)を決定する。
- ③機器制御で空調運転モード及び運転能力に基づき、室内送風機や暖房ヒーターなどの空調装置の各機器を制御する。
- ④保護機能で空調装置の各機器の状態監視を行い、故障と判定したときは、対象機器を停止する。
- ⑤故障記録で故障発生時のトラブルシューティングや空調制御状態の把握を目的としてデータを記録する。

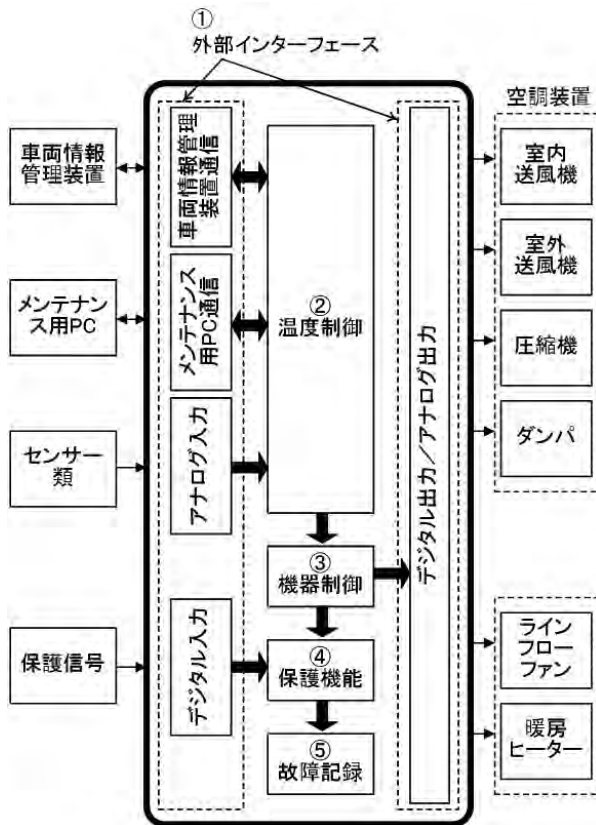


図2. 空調制御ソフトウェア機能構成図

3. 標準仕様のパターン化

これまでの鉄道会社の要求を分析し、主要機器である室内送風機、室外送風機及び圧縮機の台数の組合せにより、3パターンの標準仕様を定めた。機器台数の違いで、制御ソフトウェアに影響を与える箇所を検討した。その結果に基づき柔軟にソフトウェアを変更できる仕組みを考慮し、ソフトウェア部品化設計を行った。

3.1 空調装置の機器構成のパターン

各パターンの機器台数を表1に示す。なお、ダンパ、ラインフローファン、暖房ヒーターは主要機器ではなく、パターン化が難しい機器であるため除外した。

表1. パターンごとの機器台数

構成	室内送風機	室外送風機	圧縮機
パターンA	1	1	2
パターンB	2	2	2
パターンC	1	1	1

3.2 空調制御ソフトウェアのパターン化に向けた構成

パターンAを例にとって、空調制御ソフトウェアの主要機能である温度制御と機器制御のデータフローを図3に示す。

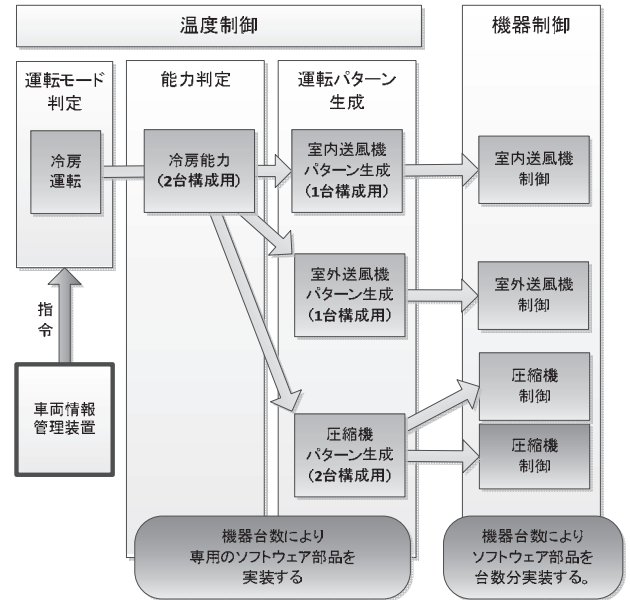


図3. データフロー

各機能の処理内容を説明する。

①運転モード判定

車両情報管理装置からの指令を受けて、冷房、暖房、自動などの運転モードを決定する。機器構成の違いによる影響はない。

②能力判定

運転モードが冷房の場合、設定温度、外気温度、車内温度、湿度、乗車率及びカレンダーにより冷房能力のレベルを決定する。冷房能力のレベルは圧縮機の台数により異なり、圧縮機2台の場合は7段階、圧縮機1台では3段階となる。機器台数の違いにより冷房能力のレベル数が異なるため機器台数ごとに処理が異なる。

③運転パターン生成

能力判定にて決定した冷房能力のレベルに従い、機器ごとのON/OFFを決定する。圧縮機については動作台数と強弱の組合せを決定する。冷房能力を下げる時、運転している2台の圧縮機から1台のみの運転に変更する場合は、故障状態や運転積算時間によりローテーション判定を行い停止する圧縮機を選択する。機器台数の違いにより運転する機器が異なるため機器台数ごとに処理が異なる。

④機器制御

運転パターン生成にて決定した機器のON/OFFと圧縮機の強弱に従い、各機器を制御する。機器台数による処理の違いはなく、機器台数分の部品を実装する。

3.3 ソフトウェア部品化設計

能力判定機能、運転パターン生成機能では、3.2で述べたように機器の台数により処理内容が大きく違ってくる。機器台数に対応した処理をそれぞれ部品化し、機器構成のパターンに応じた部品を実装することで開発の効率化を図り、機器構成の異なる車両への適用を容易に実施できるようにした。

4. モデルベース開発

モデルベース開発とは、従来、ハードウェア上でしかできなかった動作や性能の検証を、コンピュータ上でモデル化してシミュレーションを行い、製品開発の期間短縮や品質向上を図る開発手法である。今回の開発では、モデルベース開発手法を機能設計から導入した。ツールとしては、モデルベース開発のデファクトスタンダードであるMATLABを採用した。

4.1 モデルベース開発の実施方針

MATLABを用いてモデル化しコードを自動生成するモデル化部と、従来通りC言語で実装するC言語部を、図4の通り定義した。モデル化部は機器構成によって処理が異なる可能性が高い機能、C言語部はハードウェアに依存するメモリやレジスタへのアクセス処理など、モデル化できない機能と定義した。

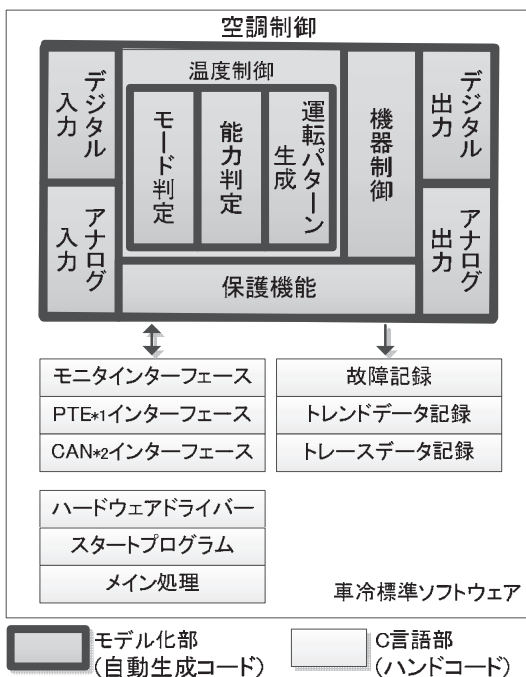


図4. モデル化適用範囲

(1) モデル化部

空調制御に関わる機能をモデル化し、インターフェース及び処理の流れの明確化を図った。従来のソフトウェア仕様書は自然言語で書かれており、ソフトウェア設計技術者の経験が浅い場合、機能設計者とソフトウェア設計者間で解釈の違いが発生する可能性がある。モデル化により、ソフトウェア設計書をブロック線図で記載することで、自然言語による仕様記述の曖昧さをなくし、仕様の認識違いを防止した。仕様記述例を図5に示す。

【自然言語での仕様記述例】

“圧縮機起動信号が入り、かつ前回の圧縮機起動信号が切りとなった場合、圧縮機起動設定を行う。”

信号入り/切りと信号ON/OFFの関係をとり間違えることによる仕様の認識違いが発生する恐れがある。

【モデルでの仕様記述例】

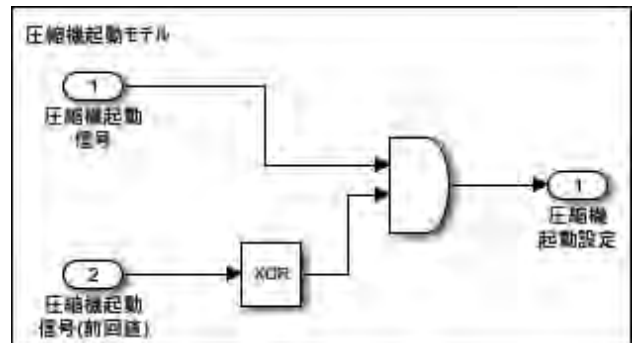


図5. 仕様記述例

(2) C言語部

標準仕様に最も近い派生ソフトウェアを流用元とし、標準化開発を行った。過去の派生ソフトウェア適用案件の障害分析と静的解析による評価を行った結果、外部変数の更新タイミングによる障害と複雑度が高い部分の障害が過半数を占めた。今回の標準化開発では、流用元プログラムからサイクロマチック複雑度の低減と外部変数の抑制を行い実装した。

さらに、C言語部の保守性を高めるため、関数のサイズ、ネスト数の見直しを実施し、表2.のように改善した。なお、各指標は静的解析ツールであるQACを用いて測定した。

表2. C言語部評価

指標	改善前	改善後
サイクロマチック複雑度(STCYC)	7.7	3.0
プログラムの難易度(STDIF)	59.8	18.9
移植時間(STPRT)	11.1	8.0

注 *1:Portable Test Equipmentの略称。

注 *2:Controller Area Networkの略称。

4.2 機能設計におけるモデルの展開

モデルは、機能ごとの入出力を明確にするトップレベルモデルを作成し、段階的に詳細化する方針で展開した。図6にモデルの詳細化イメージについて圧縮機制御機能を例に示す。

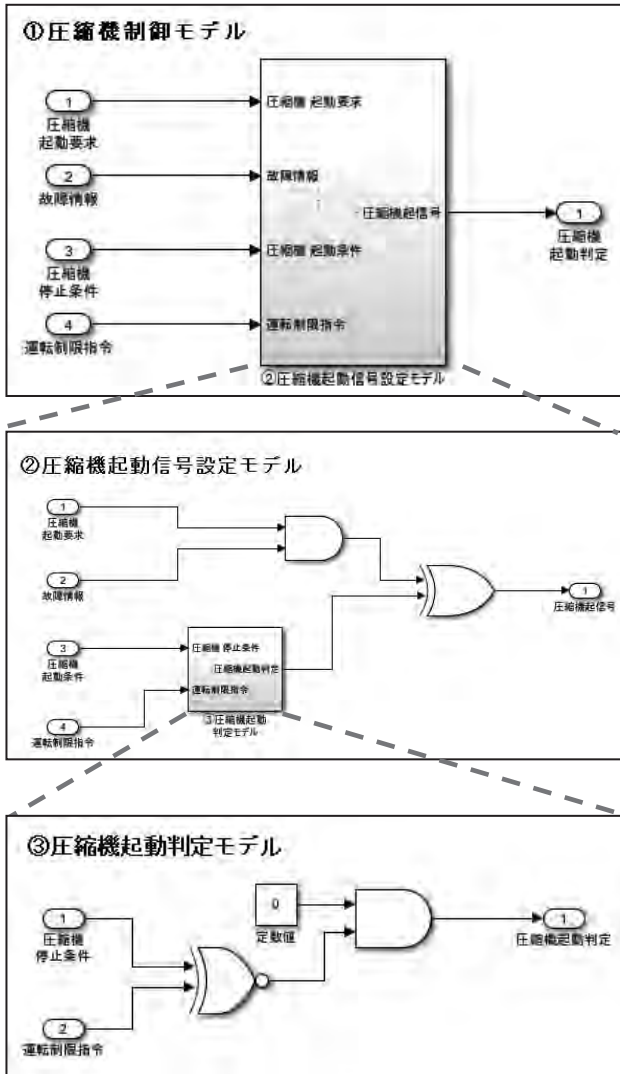


図6. モデル詳細化イメージ

- ① 圧縮機制御モデルをトップレベルモデルとして作成した。起動要求や故障情報などを入力値とし、圧縮機起動信号を出力値として設定した。これにより、圧縮機制御モデルの入出力情報を明確にした。
 - ② 圧縮機制御モデル内の処理として、圧縮機起動信号設定モデルを作成した。
 - ③ 圧縮機起動信号設定モデル内の処理として、圧縮機起動判定モデルを作成した。
- このように、処理を段階的に詳細化し、それぞれのモデルで入出力情報を明確にした。

5. モデルベース開発導入による効果

5.1 ソフトウェア設計フェーズでの改善

ソフトウェア設計段階において、MATLABを用いることで、シミュレーションによる妥当性検証を机上で行い、フロントローディング化を図った。以降に図6のモデルのシミュレーション例を示す。

- ① 圧縮機制御モデルの入力値である起動要求、故障情報、起動条件、運転制限指令の4項目の入力パターンを作成しモデルに入力する。
- ② 入力パターンの作成は、MATLABにて行い、経過時間ごとに入力値を設定する。図7に入力パターン例を示す。入力値は、「0」から開始し、4秒後に「1」、6秒後に「0」に戻す、というパターンである。

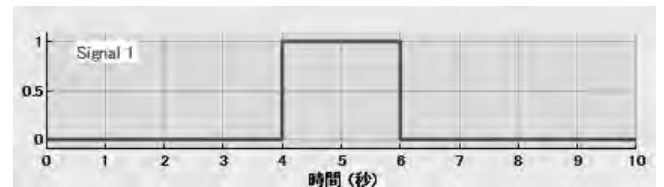


図7. 入力パターン

- ③ 作成した入力パターンをモデルに入力し、モデルでの出力結果をシミュレーションする。図8にモデルシミュレーションの例を示す。全ての入力が「1」の場合「true」が出力されることをシミュレーションしている。

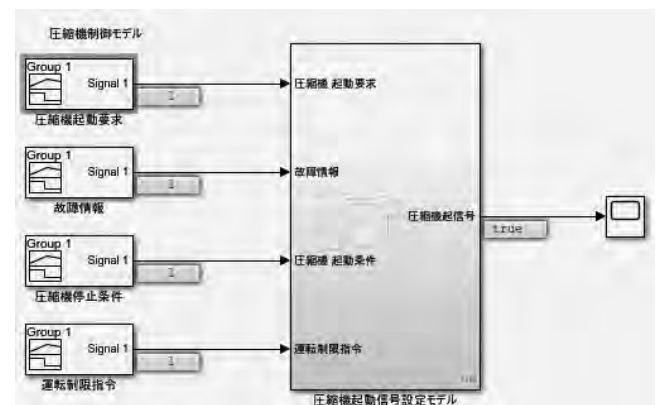


図8. モデルシミュレーション

5.2 ソフトウェア製作フェーズでの改善

従来は、ソフトウェア仕様書を基に、C言語でコーディングを行っていた。MATLABのコード自動生成機能を使用してコードの自動生成を行うことで、コーディング時間の短縮と人的なコーディングミスによる障害混入の排除を実現した。性能面でも、自動生成したコードで問題ないことを確認した。

5.3 ソフトウェア試験フェーズでの改善

ソフトウェア試験設計では、MATLABの自動パターン生成機能を使用してシミュレーションパターンを自動生成することで、効率化を図った。図8のモデルにおいて、試験パターンを自動生成した結果を図9に示す。0.2秒ごとに、各入力値のパターンを自動生成している。

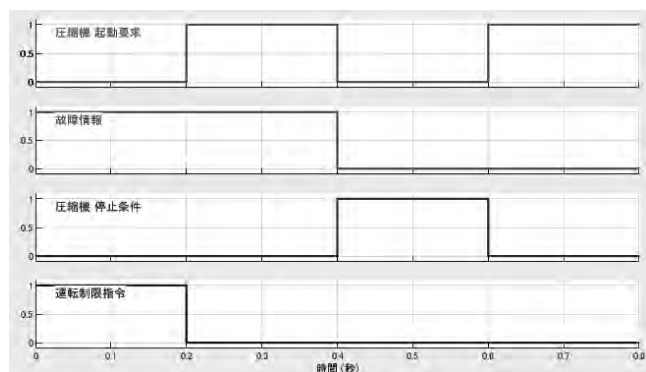


図9. 試験パターン自動生成結果

本対応により、試験項目のモレを排除できる。また、自動生成により試験設計工数を削減した。

6. 標準化開発における改善効果

今回の標準化開発により、案件展開時に見込まれる改善効果を表3に示す。効果は、ソフトウェア開発全体工数に対する工数削減率であり、開発計画の目標値を達成する見込みである。

今後は、モデルのレパートリを増やし、モデル入れ替えによる効率的な案件適用を目指す。

表3. 改善効果

フェーズ	モデル化による効果	内容	効果
設計	—	標準ソフトウェア仕様書の作成による影響範囲特定作業の効率化。	5%
	—	既存ソフトウェア構成の見直しによる流用時の設計検討時間の削減。	3%
	○	モデル化によるオーダーカスタマイズ作業の効率化。	5%
	○	設計段階でのシミュレーション検証による試験時間削減。	6%
製作	○	コード自動生成によるコーディング時間の削減。	6%
	○	コード自動生成による人的コーディングミス混入の削減による手戻り時間の削減。	
試験	○	自動試験パターン生成による試験設計作業の効率化。	5%
	○	試験項目モレの排除による再試験時間の削減。	
合計			30%

7. むすび

今回、標準ソフトウェアを開発したが、今後も環境の変化やサービス向上などにより、空調制御ソフトウェアに対する細かな要求が求められると想定される。したがって、標準ソフトウェアをベースに、柔軟にカスタマイズできる仕組みを作り、顧客の要求を満たしていく。

最後に、本開発遂行に当たり、御協力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



津田 憲宏 ツダ ノリヒロ
1985年入社。主に空調制御システムのソフトウェア開発に従事。現在、長崎事業所技術第2部車冷システム課。