

自動変速機用制御ソフトウェア開発への 実行可能な仕様書の適用に向けた取り組み

Approach to Application of Executable Specifications in Development of
Automatic Transmission Control Software

松山 芳博*
Yoshihiro MATSUYAMA

萩原 顕治*
Kenji HAGIWARA

清野 聡*
Satoshi SEINO

要 旨

ABSTRACT

自動変速機用制御ソフトウェア開発において制御信頼性向上と開発期間短縮を目的としてモデルベース開発の導入を進めている。その導入にあたって、MATLABによって記述された実行可能な仕様書、すなわちシミュレーション可能なモデルの仕様書の可読性向上と、作成時間の短縮が必要であった。そこで MAAB, J-MAAB スタイルガイドなどの一般記述ガイドラインに加えて、Stateflow を中心とした結線レス、モデル分断レス手法に基づく独自スタイルガイドの開発とドキュメント生成時におけるモデル表記の後処理により、仕様書の可読性を向上させた。さらに、そのスタイルガイドに準拠したモデリングを支援する、自動変速機制御用ブロックライブラリ、定型処理の自動モデリングツール、既存変数情報自動設定ツールを開発することにより、モデリング時間を手動モデリング時と比べて、1/20 以下に短縮することができた。その結果、自動変速機用制御ソフトウェア開発への実行可能な仕様書の適用が可能となった。

Model-based development is being introduced to the development of control software for automatic transmissions in order to increase control reliability and reduce development times. The introduction of model-based development necessitates an enhancement in the readability of executable specifications formulated using MATLAB, i.e., specifications for simulatable models, and the reduction of the time for formulation of these specifications. An original style guide was developed as a supplement to the general description guidelines provided by the MAAB and J-MAAB style guides, and the like, on the basis of a method eschewing connecting lines and division of the model based mainly on Stateflow, and post-processing was applied to the expression of the model when documents were formulated. These methods increased the readability of the specifications. In addition, a block library for automatic transmission control, an automatic modeling tool conducting routine processing, and a tool that automatically sets existing variable information were developed in order to support modeling conforming to the new style guide. These tools have reduced the time for modeling to less than one-twentieth of the time for manual modeling. This has enabled executable specifications to be employed in the development of automatic transmission control software.

1. はじめに

自動車における電子制御技術の重要性が年々高まる中、制御ソフトウェアの規模、複雑さは増加の一途をたどっている。また、制御信頼性向上、開発期間短縮、コスト削減といった要求はますます高まっている。そのような中で、車載ソフトウェア開発効率化を目的とし、近年、モデルベース開発(以下 MBD)の導入が急速に進められている^{(1), (2)}。

MBDにおいては、開発の上流である設計段階において、MATLABによって記述された実行可能な仕様書、すなわちシミュレーション可能な制御モデルが作成される。作成された制御モデルは、プラントモデルと組み合わせることで仮想制御実験が可能となる。また、実装用のCコード自動生成も可能である。

一方、MATLABは記述自由度が高く、同一の制御ロジックを記述するにしても多種多様なモデリングが可能となっ

*四輪R&Dセンター

ている。そのため可読性を低下させることにもつながり、サプライヤを含めた量産ソフトウェア開発用の制御仕様書として取り扱うことが困難であった。これに対して、自動車業界の MATLAB ユーザ団体である MAAB(The MathWorks Automotive Advisory Board), J-MAAB(Japan MATLAB Automotive Advisory Board)では、MATLAB ガイドラインを制定し、モデリング基本ラインを定義している^{(3), (4)}が、走行条件や走行環境により、変速段の切替えを多用する自動変速機制御を記述する場合、条件分岐や、それに伴う Table および Map の切替えが多数発生するため、不十分であった。そこで可読性を確保するための独自の厳密なガイドライン(以下スタイルガイド)を制定したが、可読性は向上する反面、モデリング時間が莫大となることがわかった。

2. 開発のねらい

MATLABモデルを量産ソフトウェア開発用の制御仕様書として取り扱い可能とするために、下記2項目を開発のねらいと定めた。

(1) MATLAB モデルの可読性確保

(2) MATLAB モデルのモデリング時間短縮

これらを実現するために本開発では以下を同時に開発する。

(a) 可読性の高いモデル作成スタイルガイド

(b) モデリング時間短縮化支援ツール

3. 可読性の高いモデル作成スタイルガイドの開発

スタイルガイドの開発にあたり、モデルの可読性を向上させる四つの手法を考案した。

(1) Stateflow を中心としたモデリング手法

(2) データ受け渡し用結線レス手法

(3) Table と Map 検索時のモデル分断レス手法

(4) ドキュメント生成時のモデル表記の後処理手法

以下に四つの手法の詳細について述べる。

3.1. Stateflow を中心としたモデリング手法

MATLABには大きく分けて二つの制御設計ツールが存在する。一つは Simulink で、もう一つは Stateflow である。Simulink は入力データに対して、そのデータ変換の流れを規定するデータフローの記述を得意とし、Stateflow はそのデータフローの実行順序、条件、タイミングを規定するコントロールフローの記述を得意とする。Fig. 1 と Fig. 2 にそれぞれのサンプルを示す。それぞれを制御に応じて使い分けることが可読性の高いモデリングをする上で重要となるが、自動車制御では85%以上がコントロールフローといわれている⁽⁵⁾。量産化された既存の制御仕様書を分析した結果、自動変速機用制御システムにおいても、大部分がコント

ロールフローであり、Simulink を中心としたモデリング手法では可読性が低下することがわかった。以上のことから、自動変速機用制御のモデリング手法は、Stateflow を中心とした。

3.2. データ受け渡し用結線レス手法

Simulink, Stateflow を使って制御仕様をモデリングする上で煩雑であり、かつ、可読性を低下させる要因の一つとしてデータを受け渡すための結線があげられる。ソフトウェア構造にもよるが、データの参照位置(Fig. 3の赤枠)が複数の階層にまたがって存在している場合、すべての階層に対して結線によってデータを受け渡す必要がある。モデルが大規模、かつ、多階層化し、このような変数が増加すると結線が複雑になり、可読性が低下する。

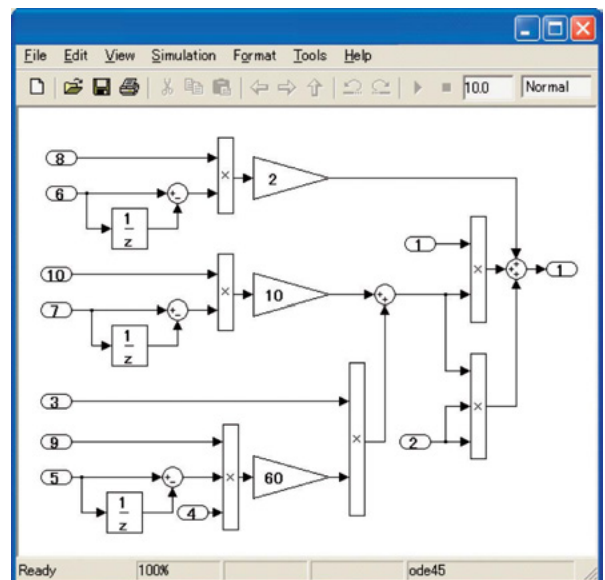


Fig. 1 Sample of Simulink model

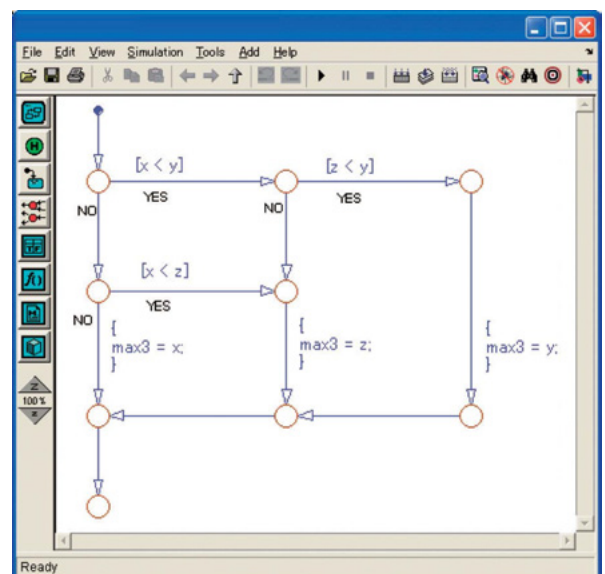


Fig. 2 Sample of Stateflow model

この課題を解決するために Stateflow 変数の変数管理手法に着目し、スタイルガイドで Stateflow チャート(以下チャート)上に記述する変数の管理則を規定した。通常、チャート上に記述する変数はすべて、マシン階層もしくはチャート階層のいずれかの階層で管理される(Fig. 4)。マシン階層で管理される変数はモデル内のすべてのチャートから、値の参照と更新が可能である。それに対して、チャート階層で管理される変数はそのチャート内でのみ、値の参照と更新が可能である。この特徴を利用し、複数のチャート間で共有する変数はマシン階層、単一のチャート内でのみ使用する変数はチャート階層でそれぞれ管理することにした。これにより複数のチャートで同一の変数を使用する場合においても、チャート間でデータを受け渡す必要がなくなり、結線不要となった。

その結果、結線の少ない可読性の高いモデル構造を実現することができた。Fig. 5 に、Fig. 3 のモデルに対して上記の変数管理手法を適用した制御モデルを示す。

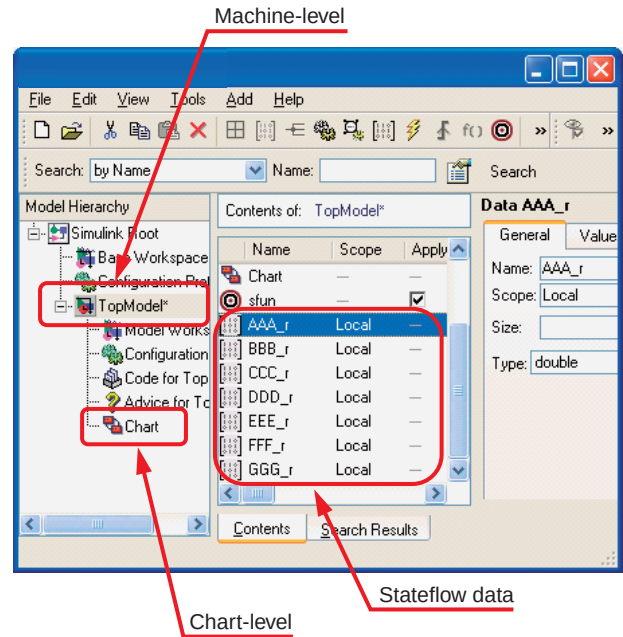


Fig. 4 Machine-level and chart-level

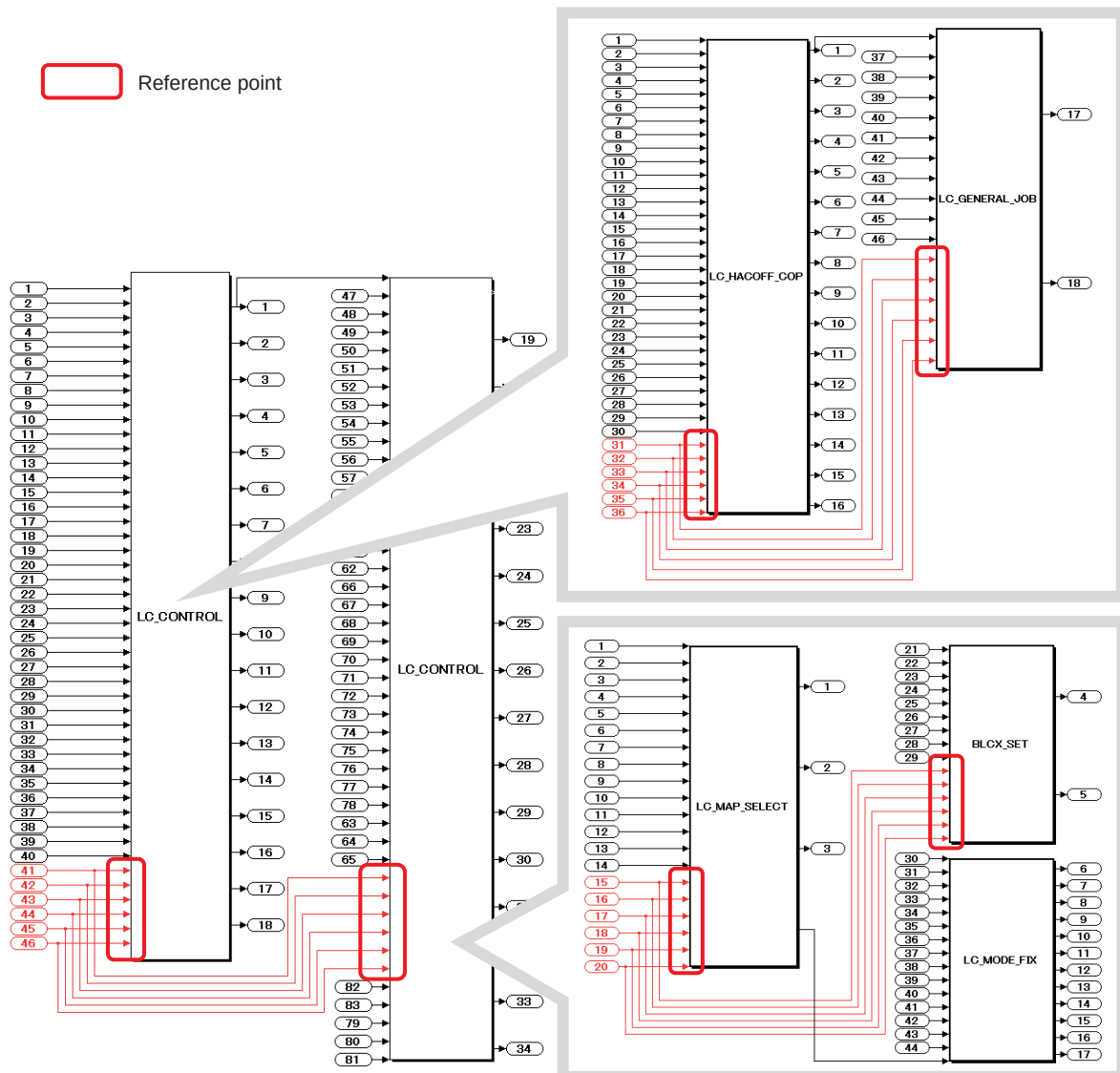


Fig. 3 Complicated model with many connecting lines

3.3. Table と Map 検索時のモデル分断レス手法

制御ロジックの中には、Fig. 6 に示す自動変速機変速特性ロジックのように、特定の引数を Table や Map と与え、その検索結果を利用して制御するものがある。この制御をモデリングするにあたり、チャート上に Table や Map 検索を直接記述することはできないため、Fig. 7 に示す Lookup Table(以下LUT)、Lookup Table(2-D)(以下LUT2D)ブロックをそれぞれ使用した。しかし、それでは Fig. 8 に示すように Table や Map 検索の前後でモデルが分断されてしまい、可読性が低下するという課題が発生した。

そこで、Fig. 9 に示すように、イベント信号と戻り値によるループ構造を用いたモデリング手法を取り入れることにした。つまり、チャート上において、Table検索をおこないたい個所で LUT ブロックを呼び出すイベント信号(Event)を出力し、それと同時に検索引数(x)を与えることでTable検索をおこない、その検索結果を再び、戻り値(y)としてチャート上に戻すという手法を考案した。これにより、モデルを分断することなく Table や Map 検索のモデリングが可能となった。

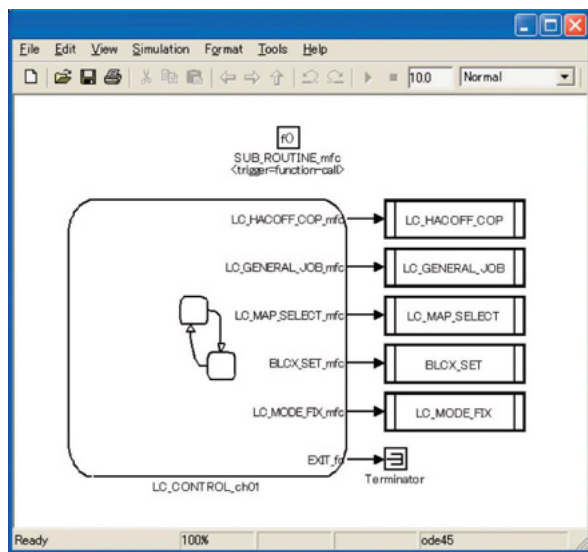


Fig. 5 Simple model with few connecting lines based on machine-level

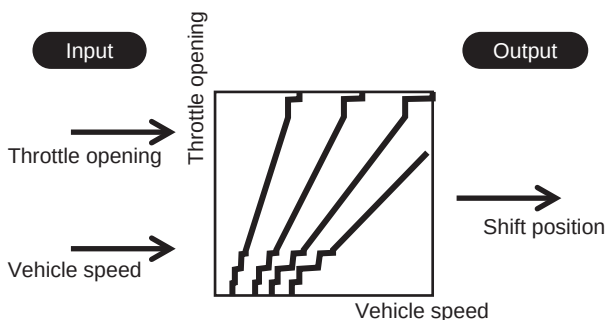


Fig. 6 Speed change characteristic logic of automatic transmission

3.4. ドキュメント生成時のモデル表記の後処理手法

量産ソフトウェア開発用の制御仕様書は、設計者だけでなく、検証者、実車およびベンチの担当者、サプライヤなど多くの開発関係者で共有される。そのため、モデルを Microsoft Office Word などの一般的なアプリケーションで閲覧可能なドキュメントにする必要があった。その際に、制御仕様書上で使用する 3 種類の変数(RAM, CALIB, WORKRAM)を視覚的に容易に識別できるように Table 1 のような表記ルールを規定した。また、条件分岐箇所はひし形表記とした。これらの表記ルールに合致したドキュメントをモデルから自動生成するために、はじめにスタイルガイドでチャート上に記述する変数の命名則(Table 2)を規定した。その上で、モデルの表記を Table 1 に合致するように後処理するツールを開発した。本ツールは MATLAB とドキュメント側のアプリケーションを連動させることで成立した。

まず、MATLAB 側で、変数の分類、条件分岐箇所の特



Fig. 7 Lookup Table and Lookup Table (2-D)

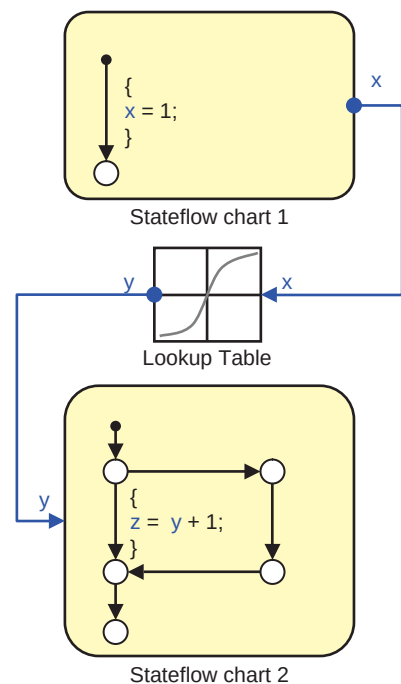


Fig. 8 Model divided into two charts by Lookup Table block

Table 1 Notation rule of variables on document

	RAM	CALIB	WORKRAM
Color	Blue	Red	Blue
Prefix	—	#	—
Capital/Small	Capital	Capital	Small

Table 2 Naming rule of variables on stateflow chart

	RAM	CALIB	WORKRAM
Suffix	_r	—	—
Capital/Small	Capital	Capital	Small

定, モデル画像のコピーをおこなうようにプログラミングした。変数の分類は, スタイルガイドで規定した命名則 (Table 2) に従っておこなった。条件分岐箇所はチャート上の Junction (Fig. 10 の丸型のオブジェクト) および Transition (Fig. 10 の矢印型のオブジェクト) の配置から特定した。Fig. 10 は, Transition 上の条件 ($X_r < Y$) が成立した時に, その Transition の矢印の方向 ($z = 1$) に遷移し, 不成立の場合には, もう一方の方向 ($z = 0$) に遷移することを表している。つまり, 条件分岐箇所では一つの Junction から二つの Transition が伸びていることがわかる。これをチャート上のすべての Junction に対して検索し, 分析することにより, 条件分岐箇所を特定した。モデル画像のコピーは, MATLAB の機能を利用し, テキスト情報を保持するために, メタ形式, すなわち画像やテキストなど複

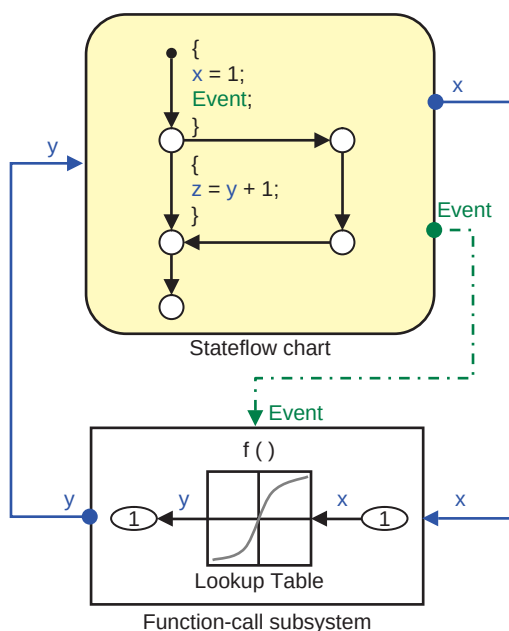


Fig. 9 Modeling method: Searching table data without division

数のタイプのデータを含むデータ形式を選択しておこなうようにした。

次に, ドキュメント側のアプリケーションを起動し, モデル画像の貼り付けと表記処理をおこなうようにプログラミングした。MATLAB からドキュメント側のアプリケーションを起動するためには, MATLABでサポートされている ActiveX 機能を利用した。また, 変数の着色やひし形図形への置換といった表記処理は, ドキュメント側のアプリケーションのオートメーション機能を利用して MATLAB 上から操作した。Fig. 11 に Fig. 10 のモデルから自動生成したドキュメントを示す。

以上の MATLAB とドキュメント側のアプリケーションを連動させたドキュメント自動生成ツールにより, 仕様書の可読性を向上させることができた。

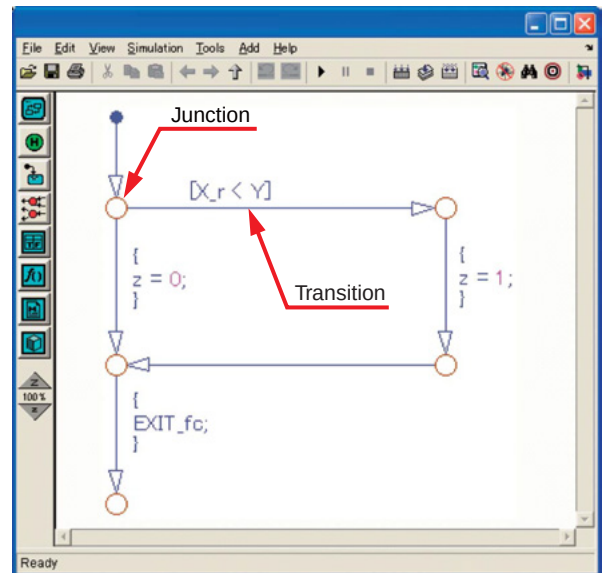


Fig. 10 Junction and transition

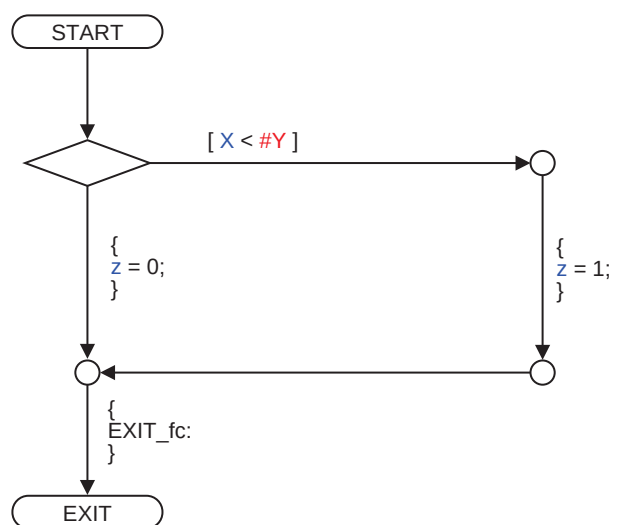


Fig. 11 Document generated from model in Fig. 10

4. モデリング時間短縮化支援ツールの開発

第3章では、スタイルガイドの開発にあたり、モデルの可読性向上をねらいとして考案した四つの手法について述べた。本章では、そのスタイルガイドに準拠した実行可能な仕様書のモデリング時間短縮をねらいとして開発した下記三つのモデリング支援ツールについて述べる。

- (1) 自動変速機制御用ブロックライブラリ
- (2) 定型処理の自動モデリングツール
- (3) 既存変数情報の自動設定ツール

4.1. 自動変速機制御用ブロックライブラリ

MATLAB 標準のブロックライブラリには、Fig. 12 に示すように、多種多様な制御設計用ブロックが用意されている。その中から、自動変速機制御をスタイルガイドに準拠してモデリングする際に必要となるブロックを集め、カスタマイズした独自のブロックライブラリを開発した(Fig. 13)。本ライブラリのブロックは、ブロックに設定された情報をマスク上に自動表示するようにカスタマイズした。その具体例を Fig. 14 に示す。Fig. 14 の Table 検索ブロックは Graphical User Interface (以下 GUI)に軸データ名を入力すると、その情報がマスク上に自動表示される。

この仕組みは、MATLAB の GUI 開発環境である GUIDE を利用して、軸データ名を入力するための GUI と、その入力情報をマスク上に自動表示するための M-File プログラムを作成することにより実現した。

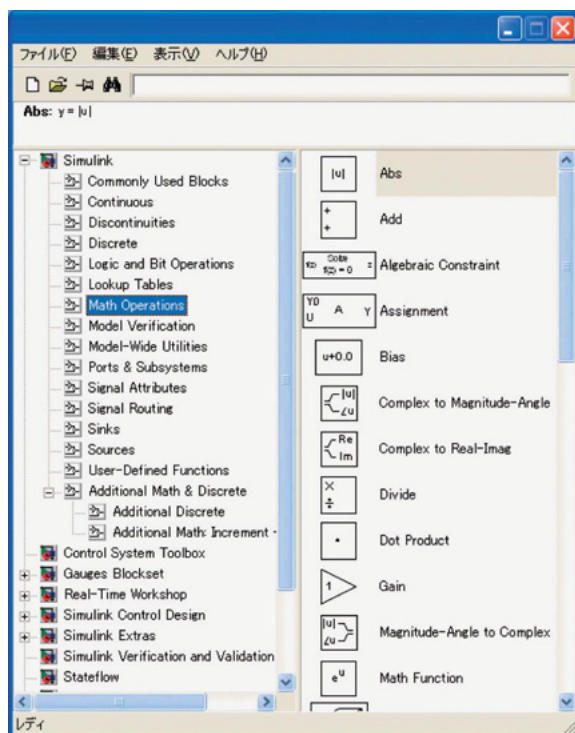


Fig. 12 Simulink library

以上の自動変速機制御用ブロックライブラリによりモデルの可読性向上とモデリング時間の短縮が可能となった。

4.2. 定型処理の自動モデリングツール

第3章で述べた、Table 検索や、Map 検索を適用した制御ロジックでは、複数の Table や Map を制御状態に応じて切替えて使用する場合がある。その場合、モデルは、Fig. 15 に示すような Switch Case ブロック、Merge ブロック、さらに、4.1 節で述べたカスタマイズした Table または Map ブロックなどから構成される定型的なものとなる。しかし、切替え数が多ければ多いほど、単純なモデリング作業が増大し、モデリングエラーも増加する。

そこで、このような定型的なモデリングを自動で実行可能とする自動モデリングツールを開発した。本ツールは、設計者の情報入力用 GUI と、自動モデリングプログラムから構成した。

まず、GUI の設計にあたって、量産化された既存の自動

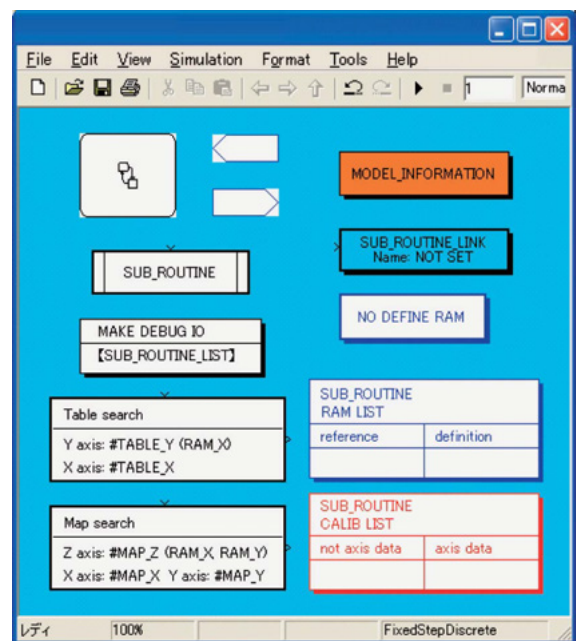


Fig. 13 Automatic transmission modeling library

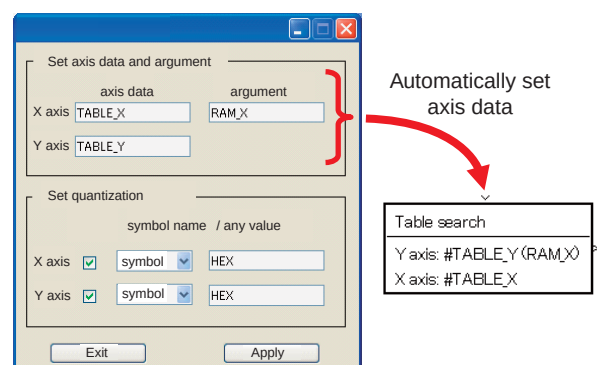


Fig. 14 GUI and customized table block

変速機用制御仕様書において、切替え制御パターンの分析をおこなった。その結果、TableとMapの切替え以外に、CALIBデータの切替え、RAMデータの切替え、さらには多条件でのMapの切替えなどが代表的な五つの切替えパターンであることがわかった。また、その切替え数や、切替え条件は、制御に応じてさまざまであり、高い設計自由度が要求されることがわかった。そこで、Fig. 16 に示すように、GUI構成は、設計者が五つのパターンを自由に選択でき、さらに、切替え数や切替え条件を容易に設定できるものとした。

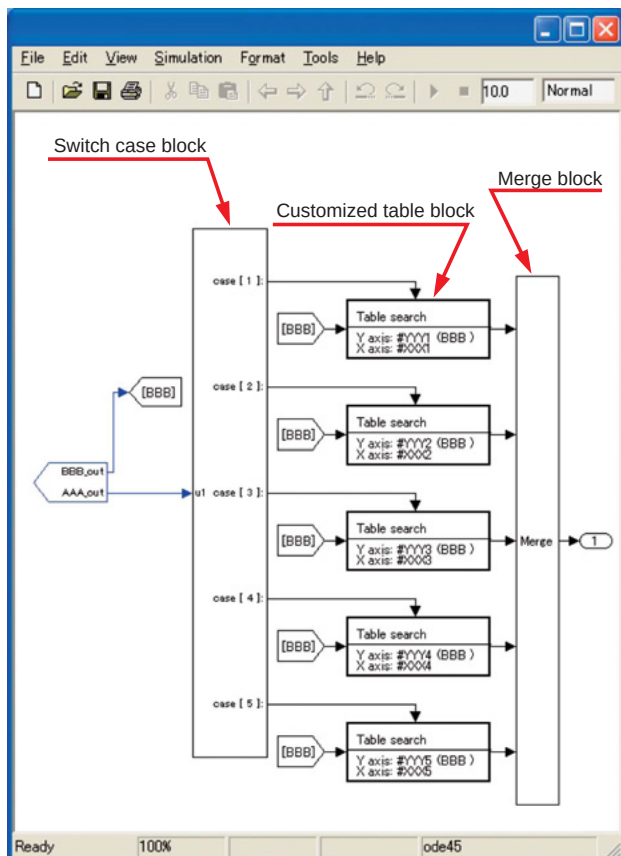


Fig. 15 Table switching model

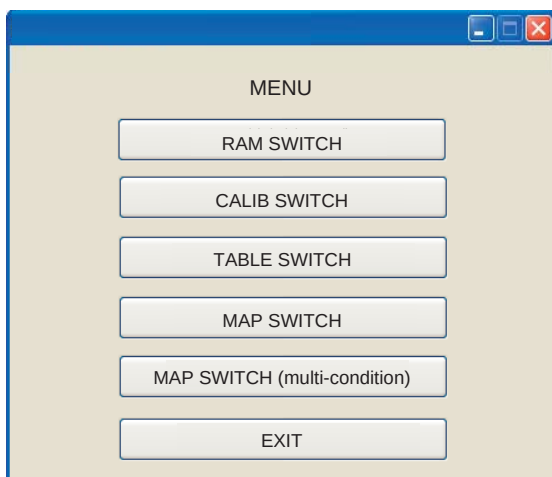


Fig. 16 GUI of auto-modeling tool

次に、GUIによって選択し、設定された情報に基づき自動的に Simulink や Stateflow によってモデリングするプログラムを作成した。そのプログラムは、下記一連の処理を自動的にこなうものとした。

- (1) GUIで選択された切替えパターンに対し、自動モデリングに必要なブロックを 4.1. 節のブロックライブラリから取得し、Function-Call Subsystem 内に配置する。
- (2) 配置したブロックに対し、GUIで設定した情報を反映する。
- (3) ブロック同士を結線する。
- (4) Function-Call Subsystem のマスク上に切替えロジックを表記する。

Fig. 17 に本ツールによる自動モデリング結果を示す。

以上のGUIと自動モデリングプログラムから構成された自動モデリングツールにより、モデリング時間の短縮が可能となった。

4.3. 既存変数情報の自動設定ツール

量産ソフトウェア開発用の制御仕様書では、その中で取り扱われるすべての変数に対して、データスケール(LSB)、データタイプといった変数の精度を指定するための情報を設定する必要がある。MATLABモデルを量産の制御仕様書として取り扱う場合も同様の設定が必要となる。ここで、大規模な制御モデルにおいて、さらに莫大な数の変数に対して一変数ごとに情報設定するには、莫大な時間を要する。

一方、量産ソフトウェアで取り扱われている既存の変数については、その情報含めて、データベースで管理されており再利用が容易となっている。

そこで、データベースとリンクし、既存変数情報を自動取得、一括設定し、その設定結果を一覧表示するGUIを開発した。さらに、その変数の情報を MATLAB モデル内に保持するための専用ブロックを開発した。

データベースとのリンク、すなわち、既存変数情報の自

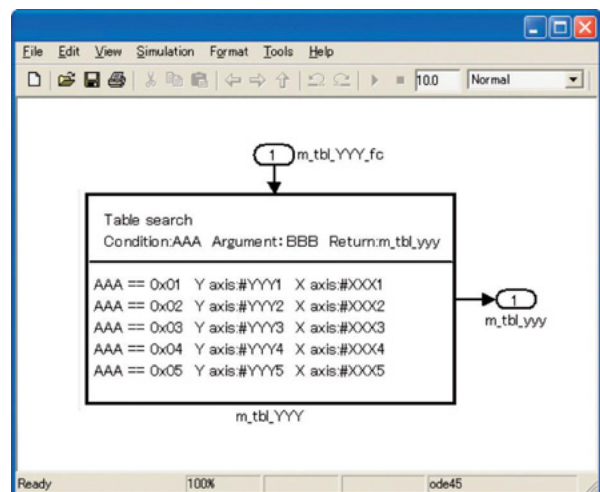


Fig. 17 Result of auto-modeling

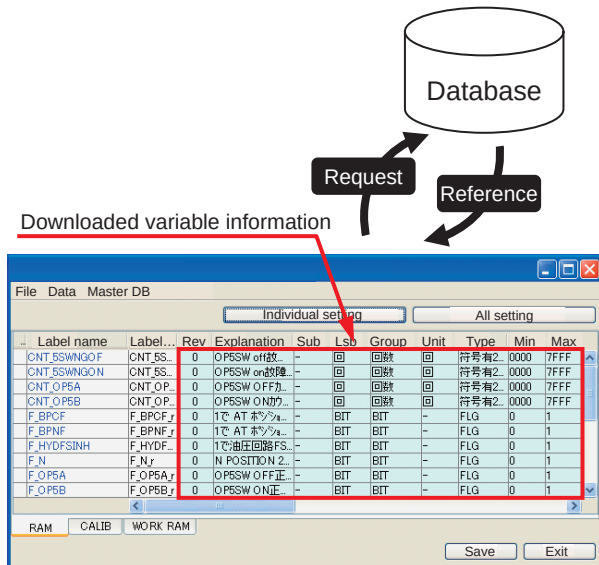


Fig. 18 Link system between database and GUI managing variable information

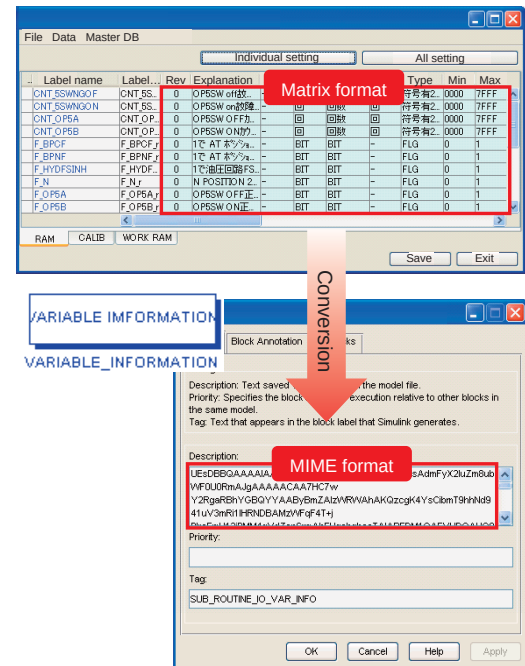


Fig. 19 Conversion of variable information format

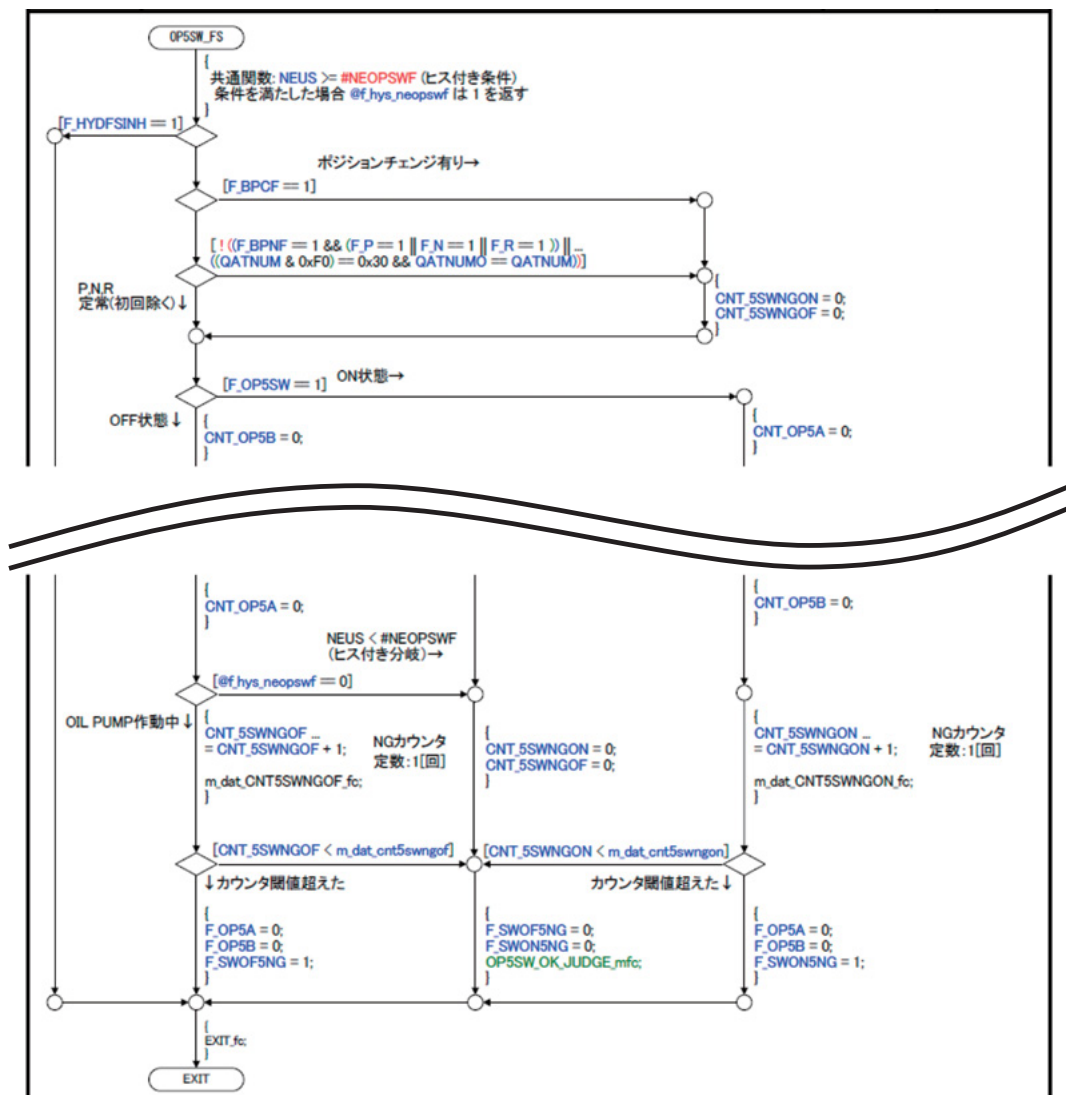


Fig. 20 Example of application

動取得、一括設定にあたっては、まず、GUI 起動時に、スタイルガイドで規定した命名則 (Table 2) に従ってモデルを自動解析し、設定が必要な変数をその変数名から自動抽出できるようにプログラミングした。その上で、データベース言語の一つである SQL を使用して、データベースから、変数名に関連付けされた変数情報を参照できるようにした (Fig. 18)。

一括設定された変数情報の MATLAB モデル内での保持にあたっては、変数情報を専用ブロックのプロパティ欄に格納できるようにプログラミングした。格納する際のデータ形式として電子メールなどで利用される MIME 形式を採用することにより、複数の変数情報から構成される行列データを一つのブロックで保持できるようにした (Fig. 19)。

以上の既存変数情報の自動設定ツールにより、モデリング時間の短縮が可能となった。

5. 適用事例

開発したスタイルガイド、モデリング支援ツールを適用した自動変速機用制御ソフトウェアの実行可能な仕様書を Fig. 20 に示す。量産ソフトウェア開発用の制御仕様書として取り扱い可能な MATLAB モデルを、ツール導入前と比較して 1/20 以下の時間で作成できるようになった。

6. まとめ

- (1) Stateflow を中心とした結線レス、モデル分断レス手法に基づくスタイルガイドの開発と、ドキュメント生成時におけるモデル表記の後処理ツールの開発により自動変速機用制御仕様書の可読性を向上させることができた。
- (2) 自動変速機制御用ブロックライブラリ、定型処理の自動モデリングツール、既存変数情報の自動設定ツールを開発することにより、上記スタイルガイドに準拠したモデリング時間を 1/20 以下に短縮することができた。

その結果、自動変速機用制御ソフトウェア開発への実行可能な仕様書の適用が可能となった。

参考文献

- (1) 大塚郁, 大井正也: 車載ソフトウェア開発へのモデルベース開発適用, 自動車技術, Vol.60, No.6, p.80-8 (2006)
- (2) 箕輪信太郎, 廣田俊明, 横山浩紀, 石黒哲矢, 大塚訓: パワープラント制御ソフトの自動生成, Honda R&D Technical Review, Vol.16, No.1, p.243-248
- (3) <http://www.mathworks.com/industries/auto/maab.html>

(4) <http://j-maab.cybernet.jp/>

(5) http://www.cybernet.co.jp/documents/pdf/products/magazine/cybernet_news/no87.pdf

■著 者■



松山 芳博



萩原 顕治



清野 聡