

状態遷移や条件分岐を含む制御モデルを対象とした Simulink・UMLモデル変換ツール

黒木 裕太¹ 田中 亨祐¹ 兪 明連¹ 横山 孝典¹

概要: 本論文では、組み込み制御ソフトウェアの開発効率向上を目的に、状態遷移や条件分岐を含む制御モデルに対応可能な Simulink・UML モデル変換ツールを提案する。Simulink モデルにおいて状態遷移や条件分岐を表すために用いられるブロックとして、Stateflow ブロック、Switch ブロック、MultiportSwitch ブロック、If ブロック、Switch Case ブロック、Merge ブロックがある。本モデル変換ツールは状態遷移や条件分岐を含む Simulink モデルを入力とし、状態遷移や条件分岐に応じて実行する処理を切り替えられる UML モデル、具体的にはクラス図、オブジェクト図、シーケンス図、およびアクティビティ図を出力する。そして、複数の Simulink モデルに対して適用実験を行い、本モデル変換ツールの有用性を確認した。

キーワード: 組み込みソフトウェア、モデルベース開発、UML、モデル変換

1. はじめに

自動車や家電製品に用いられる組み込み制御ソフトウェアの開発量は増大しており、開発効率の向上が重要な課題となっている。開発効率を向上させる手法として、MATLAB/Simulink[1] 等の制御系 CAD/CAE ツールを用いたモデルベース開発が注目されている。しかし、制御モデル設計用のツールは制御ロジックの設計には適しているが、ソフトウェア設計に十分な機能を提供していないと言われており、Sangiovanni-Vincentelli らは、制御系 CAD/CAE ツールでは機能モデルとアーキテクチャモデルを分離して記述することができない、タスクやリソース等のモデルが定義されていない等の問題を指摘している [2]。そのため、制御系 CAD/CAE ツールは制御設計にのみ用いて、ソフトウェア設計には UML のようなソフトウェアを対象としたモデリング言語を用いるべきと考える。そのためには、制御系 CAD/CAE ツールを用いて作成した制御モデルをソフトウェア設計に適した UML モデルに効率よく変換できるツールが必要である。

Simulink モデルから UML モデルへの変換については、既に様々な変換ツールが提案されている。Ramos-Hernandez らは、Simulink モデルを UML モデルのクラス図に自動変換するツールを開発している [3], [4]。このツールは Simulink モデルの各ブロックをインタフェースクラスに、各ライン

を依存関係に変換する。Müller-Glaser らは、Simulink モデルを UML モデルのクラス図とコミュニケーション図に自動変換するツールを開発している [5], [6]。このツールは Simulink モデルの各ブロック、ライン、分岐をクラスに変換してクラス図を出力する。そして、各クラスをオブジェクトに変換することでコミュニケーション図を出力する。Sjöstedt らは、Simulink モデルを UML モデルの複合構造図とアクティビティ図に自動変換するツールを開発している [7]。アクティビティ図変換では、Simulink モデルのシミュレーション実行順序をアクティビティ図として出力している。しかし、これらは Simulink モデルの要素ひとつひとつを UML モデルの要素に変換しているため、必ずしも再利用性の高い UML モデルとはならない。

そこで我々は、制御上重要な意味を持つデータに着目して階層化した Simulink モデルから、再利用性の高いクラスから成る UML モデルを生成するモデル変換ツールを提案した [8]。本モデル変換ツールは Subsystem ブロックから成る Simulink モデルを UML モデルのクラス図、オブジェクト図、シーケンス図、アクティビティ図に変換する。しかし、状態遷移や条件分岐を含む Simulink モデルは対象としていなかった。状態遷移や条件分岐を表現するブロックを Subsystem 化すれば変換は可能であるが、実行効率の良い UML モデルとなってしまう。

本研究の目的は、これまで扱っていなかった状態遷移や条件分岐を含む Simulink モデルを入力することで、制御フローを考慮した実行効率の良い UML モデルを生成可能

¹ 東京都市大学
Tokyo City University
{g1481519, g1581515, myoo, tyoko}@tcu.ac.jp

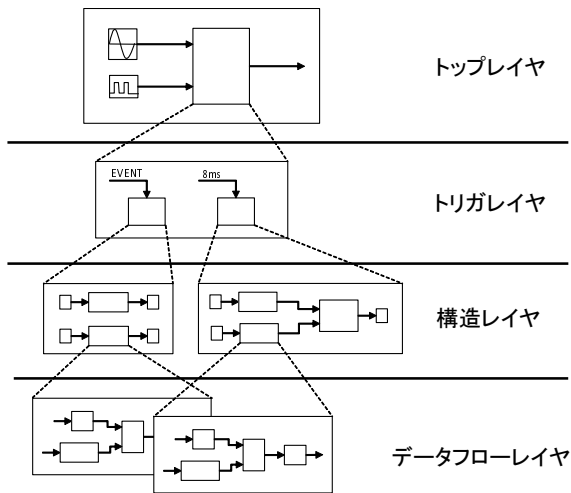


図 1 MAAB の階層構造

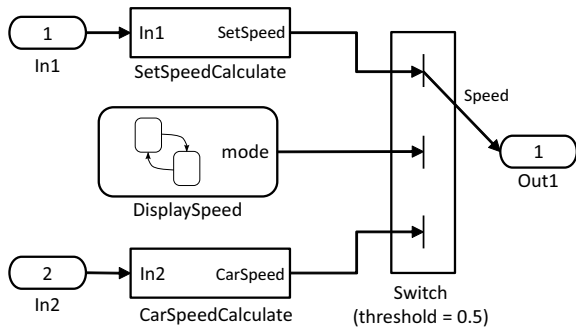


図 2 Switch ブロックを用いた Simulink モデルの例

なモデル変換ツールを開発することである。

以下本論文では、まず 2 章で本研究で変換対象とする Simulink モデルについて述べる。続いて、3 章で本研究で提案するモデル変換方法について説明する。さらに、4 章でモデル変換ツールの実装、5 章で適用実験について述べ、6 章で関連研究との比較を行う。

2. 変換対象のモデル

2.1 変換対象範囲

本研究では Simulink モデルは、MATLAB のユーザ団体である MathWorks Automotive Advisory Board (以下、MAAB) が制定した制御モデル記述のためのガイドライン [9] で規定された階層における構造レイヤのモデルを変換対象とする。図 1 に示すように MAAB の階層構造は、トップレイヤ、トリガレイヤ、構造レイヤ、データフローレイヤの 4 階層から構成される。トップレイヤでは、モデルの入出力を記述し、トリガレイヤでは演算のタイミングを記述する。構造レイヤでは、制御ロジックの構造を記述し、データフローレイヤではデータの流れを記述する。我々は、制御上重要なデータのみを構造レイヤに現れるようにすることで再利用性に適したクラス構成に変換可能とする [8]。構造レイヤで使用可能なブロックには Subsystem ブロックのほか、全ての階層で使用できるブロックとして

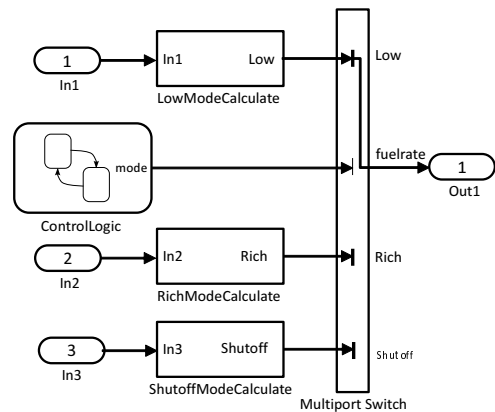


図 3 MultiportSwitch ブロックを用いた Simulink モデルの例

SwitchCase ブロックなどガイドライン Version 3.0 では 23 種類、Version 2.0 では 22 種類のブロックが使用できる。

我々は、それらのうち状態遷移や条件分岐を含む制御モデルを表現するために用いられる可能性のある Inport ブロック、Outport ブロック、Subsystem ブロック、Stateflow ブロック、Switch ブロック、MultiportSwitch ブロック、If ブロック、SwitchCase ブロック、Merge ブロックの 9 種類のブロックを対象とする。Switch ブロック及び MultiportSwitch ブロックは Version 2.0 では使用可能であったが、Version 3.0 からは除かれている。しかし、現実には広く使用されているため対象に含めている。なお、最近公開されたガイドライン Version 4.0 [10] ではモデルの階層構造はルールではなく考え方になっているが、サブ機能レイヤ、制御レイヤ、選択レイヤが変換対象の階層に対応すると考えている。

2.2 対象とする Simulink モデルのパターン

以上のブロックを用いて構成される状態遷移や条件分岐を含む Simulink モデルは、Switch ブロックを用いた Simulink モデル、MultiportSwitch ブロックを用いた Simulink モデル、If ブロックを用いた Simulink モデル、および SwitchCase ブロックを用いた Simulink モデルの 4 種類のパターンに分類できる。以下、それら 4 種類の変換対象の Simulink モデルについて説明する。

Switch ブロックを用いた Simulink モデルの例を図 2 に示す。この Simulink モデルは、Stateflow ブロックである表示速度指定 (DisplaySpeed) が出力する選択信号により Subsystem ブロックである設定速度算出 (SetSpeedCalculate) のデータまたは車速算出 (CarSpeedCalculate) のデータ的一方を Out1 ブロックに出力するモデルである。mode がしきい値 (threshold) の 0.5 よりも大きい場合は、設定速度算出のデータを Out1 に出力する。また、mode がしきい値 0.5 よりも小さい場合は、車速算出のデータを Out1 に出力する。

MultiportSwitch ブロックを用いた Simulink モデルの

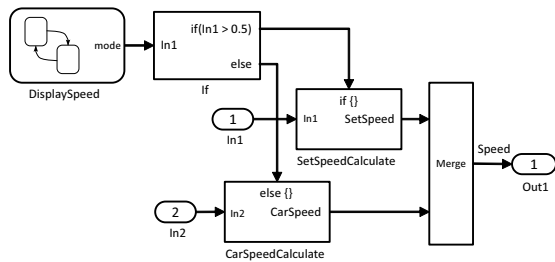


図 4 If ブロックを用いた Simulink モデルの例

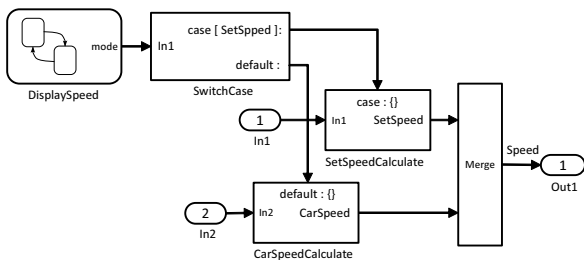


図 5 SwitchCase ブロックを用いた Simulink モデルの例

例を図 3 に示す。この Simulink モデルは、Stateflow ブロックである制御ロジック (ControlLogic) が出力する選択信号により Subsystem ブロックである Low モード算出 (LowModeCalculate) または Rich モード算出 (RichModeCalculate), Shutoff モード算出 (ShutoffModeCalculate) のうち 1 つのデータを Out1 ブロックに出力するモデルである。mode が Low であった場合は、Low モード算出のデータを Out1 に出力する。また、mode が Rich であった場合は Rich モード算出のデータを、mode が Shutoff であった場合は Shutoff モード算出のデータを Out1 へ出力する。

If ブロックを用いた Simulink モデルは、If ブロックとともに Merge ブロックを使用して条件分岐を表現する。図 4 は、図 2 の Simulink モデルを If ブロックを用いて表現した Simulink モデルの例である。この Simulink モデルは、Stateflow ブロックである表示速度指定が出力する選択信号をもとに If ブロックが持つ条件式によって算出する Subsystem ブロック (設定速度算出または車速算出) を切り替えるモデルである。mode が 0.5 よりも大きい場合は、設定速度算出のデータを Out1 に出力する。また、mode が 0.5 よりも小さい場合は、車速算出のデータを Out1 に出力する。

SwitchCase ブロックを用いた Simulink モデルも Switch-Case ブロックとともに Merge ブロックを使用する。図 5 は、図 2 の Simulink モデルを SwitchCase ブロックを用いて表現した Simulink モデルの例である。この Simulink モデルは、Stateflow ブロックである表示速度指定が出力する選択信号をもとに SwitchCase ブロックが持つ条件式によって算出する Subsystem ブロック (設定速度算出または車速算出) を切り替えるモデルである。mode が設定速度である場合は、設定速度算出のデータを Out1 に出力す

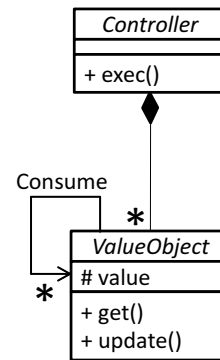


図 6 アーキテクチャパターン

る。また、mode が車速である場合は、車速算出のデータを Out1 に出力する。

If ブロックや SwitchCase ブロックを用いた Simulink モデルは制御フローがあきらかに記述されているため、条件分岐をともなうソフトウェアモデルとしてそのまま変換できるが、Switch ブロックや MultiportSwitch ブロックを用いた Simulink モデルは、データフローを解析して制御フローとして条件分岐を持つソフトウェアモデルに変換する必要がある。

2.3 生成する UML モデルのクラス構成

Simulink モデルを変換する際は、図 6 で表現されるアーキテクチャパターンに基づく UML モデルに変換する。図 6 に示すように、本アーキテクチャパターンは、Simulink モデル全体を表す抽象クラスであるコントローラ (Controller) クラス、各ブロックに対応したクラスの抽象クラスであるデータ値オブジェクト (ValueObject) クラスから成る。コントローラクラスは、各ブロックに対応するクラスの update メソッドを呼び出すための exec メソッドを持つ。データ値オブジェクトクラスは、内部データを表す変数 value とデータの読み出しを行う get メソッドと、データの更新を行う update メソッドを持つ。

変換に際しては、まず、図 6 のコントローラクラスとデータ値オブジェクトクラスを生成する。コントローラクラスとデータ値オブジェクトクラス間はコンポジションの関係として生成し、データ値オブジェクトクラスはブロックに対応したクラス間で関連を持つため Consume 関連を生成する。そして、各ブロックに対応するクラスは、データ値オブジェクトクラスのサブクラスとして表現する。

3. 変換方法

3.1 変換方針

本研究で変換対象とする 4 種類の Simulink モデルは、それぞれ異なる方法で変換を行う必要がある。

また、Switch ブロックを用いた Simulink モデルにおいて、選択する Subsystem ブロックの出力データ名が異なる

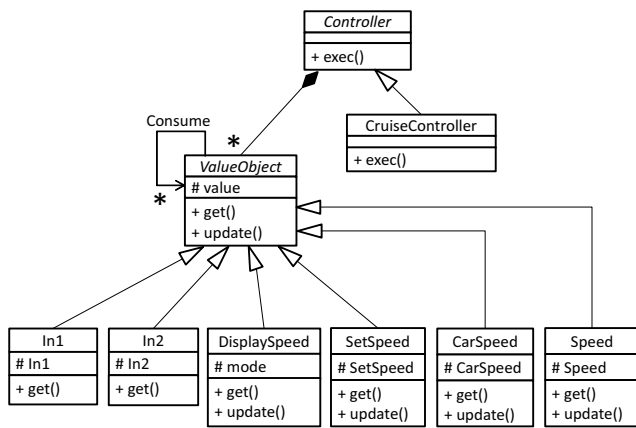


図 7 図 2 の Simulink モデルに対応するクラス図

場合と同じ場合について別々の変換方法を提案する。出力データ名が異なる場合は 2 つの Subsystem ブロックを別のデータを算出するものとして設計されたと考えられるため、2 つの Subsystem ブロックと Switch ブロックを別々のクラスと見なして変換を行う。また、出力データ名が同じ場合はひとつのデータを算出するものとして設計されたと考えられるため、2 つの Subsystem ブロックと Switch ブロックを 1 つのクラスと見なして変換を行う。

図 3 の MultiportSwitch ブロックを用いた Simulink モデルにおいても同様に、選択する Subsystem ブロックの出力データ名が異なる場合と同じ場合について別々の変換方法を提案する。出力データ名が全て異なる場合は、各 Subsystem ブロックと MultiportSwitch ブロックを別々のクラスと見なして変換を行う。また、出力データ名が全て同じ場合は、各 Subsystem ブロックと MultiportSwitch ブロックを 1 つのクラスと見なして変換を行う。

図 4 の If ブロックを用いた Simulink モデルは、If ブロックと Merge ブロックにより条件分岐を記述しているため、それらをまとめて 1 つのクラスと見なし、変換を行う。この場合も、選択する Subsystem ブロックの出力データ名が異なる場合と同じ場合について別々の変換方法を提案する。出力データ名が異なる場合は、If ブロックと Merge ブロック、2 つの Subsystem ブロックの 3 つを別々のクラスと見なして変換を行う。また、出力データ名が同じ場合は、If ブロックと Merge ブロック、2 つの Subsystem ブロックの 3 つを 1 つのクラスと見なして変換を行う。図 5 の SwitchCase ブロックを Simulink モデルにおいても同様の考え方で変換する。

以下、3.2 節、3.3 節で、データフローを解析して条件分岐を持つ制御フローに表現する必要がある Switch ブロックを用いた Simulink モデルの場合について具体的な変換方法を説明する。

3.2 選択する出力データ名が異なる場合

図 2 の Simulink モデルは、2 つの Subsystem ブロック

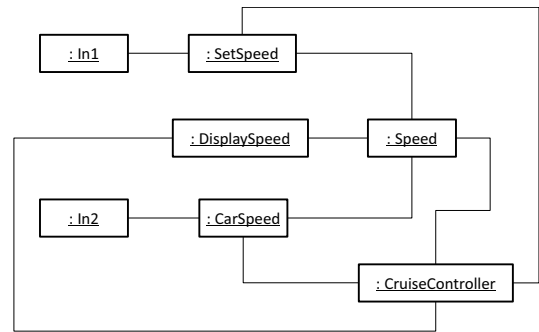


図 8 図 2 の Simulink モデルに対応するオブジェクト図

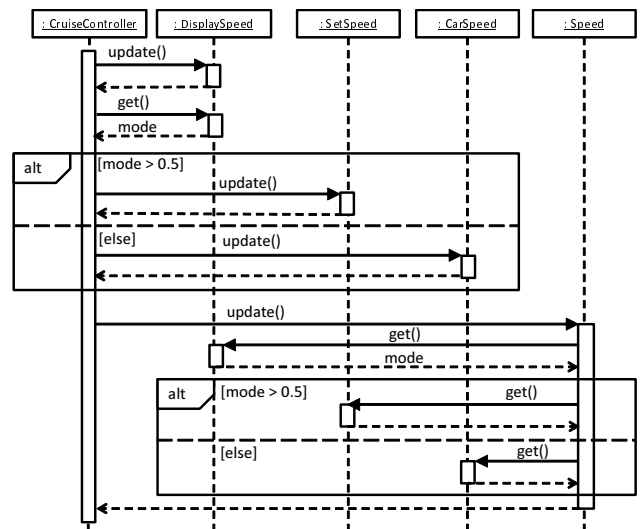


図 9 図 2 の Simulink モデルに対応するシーケンス図

の出力データ名が設定速度 (SetSpeed) と車速 (CarSpeed) のように異なるため、設定速度、車速、速度をそれぞれ別々のクラスと見なして変換を行う。例えば、図 2 の Simulink モデルを UML モデルに変換する場合、図 7 に示すクラス図、図 8 に示すオブジェクト図、図 9 に示すシーケンス図を出力する。

図 7 のクラス図に示すように、In1, In2, 設定速度、表示速度指定、車速、速度 (Speed) に対応したクラスは、データ値オブジェクトクラスのサブクラスとして生成する。Inport ブロックである In1, In2 に対応するクラスは、内部データを表す変数とデータの読み出しを行う get メソッドのみ持つ。設定速度、表示速度指定、車速、速度に対応するクラスは、内部データを表す変数とデータの読み出しを行う get メソッドと、データの更新を行う update メソッドの両方を持つ。また、コントローラ抽象クラスのサブクラスとしてクルーズ制御 (CruiseController) クラスを生成する。クルーズ制御クラスは、設定速度クラス、表示速度指定クラス、車速クラス、速度クラスの update メソッドを呼び出すための exec メソッドを持つ。

図 8 のオブジェクト図に示すように、図 7 で出力したクラス図の In1 クラス、In2 クラス、設定速度クラス、表示速度指定クラス、車速クラス、速度クラス、クルーズ制御

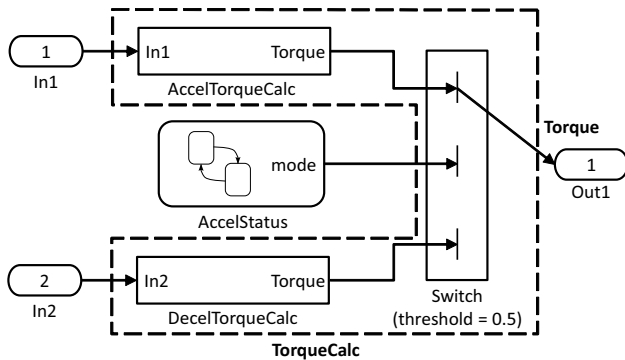


図 10 トルク算出の Simulink モデル

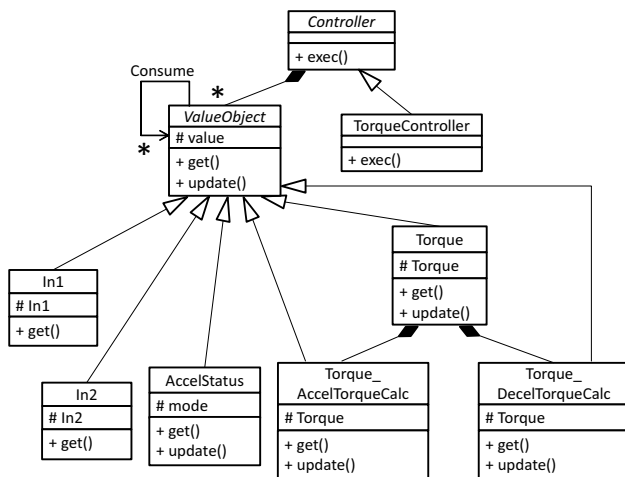


図 11 図 10 の Simulink モデルに対応するクラス図

クラスに対してオブジェクトを生成する。また、オブジェクト間のリンクは Simulink モデルのラインを基に生成する。最後に、update メソッドを持つオブジェクトである設定速度、表示速度指定、車速、速度とクルーズ制御間のリンクを生成する。

図 9 のシーケンス図に示すように、クルーズ制御クラスが表示速度指定クラスの update メソッドを呼び出して状態値を更新し、get メソッドを呼び出して状態値を取得する。そして、条件式に基づいて実行する処理を切り替える。状態値 (mode) がしきい値 (threshold)0.5 よりも大きい場合は、設定速度クラスの update メソッドを呼び出して処理を実行させる。状態値がしきい値 0.5 よりも小さい場合は、車速クラスの update メソッドを呼び出して処理を実行させる。最後に、速度クラスの update メソッドを呼び出し表示速度指定クラスの get メソッドを呼び出し状態値を取得して、実行したクラスの出力データを取得する。

If ブロックや SwitchCase ブロックを用いた Simulink モデルにおいても、3.1 節の考え方をを用いることで図 7~図 9 の UML モデルに変換することができる。

3.3 選択する出力データ名が同じ場合

図 10 の Simulink モデルは、2 つの Subsystem ブロッ

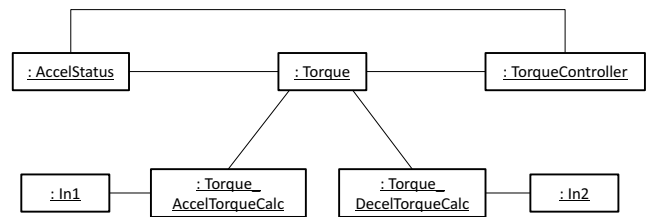


図 12 図 10 の Simulink モデルに対応するオブジェクト図

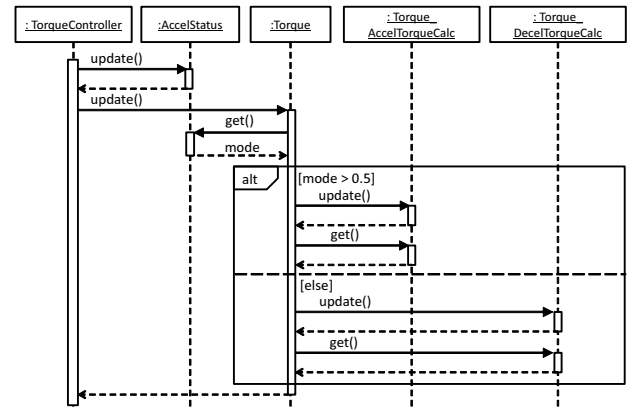


図 13 図 10 の Simulink モデルに対応するシーケンス図

クの出力データ名がトルク (Torque) と同じであるため、加速時トルク算出 (AccelTorqueCalc)、減速時トルク算出 (DecelTorqueCalc) および Switch ブロックを 1 つのクラスと見なして変換する。例えば、図 10 の Simulink モデルを UML モデルに変換する場合、図 11 に示すクラス図、図 12 に示すオブジェクト図、図 13 に示すシーケンス図を出力する。

図 11 のクラス図に示すように、In1 クラス、In2 クラス、加減速状態 (AccelStatus) クラスは、データ値オブジェクトクラスのサブクラスとして生成する。また、コントローラ抽象クラスのサブクラスとしてトルク制御 (TorqueController) クラスを生成する。そして、加速時トルク算出、減速時トルク算出および Switch ブロックをまとめてトルクを表すクラスとし、その下位クラスとして加速時トルク算出クラスおよび減速時トルク算出クラスを生成する。上位クラスと下位クラスの間をコンポジションの関係として生成する。

図 12 のオブジェクト図に示すように、図 11 で出力したクラス図のトルク制御クラス、トルククラス、加減速状態クラス、In1 クラス、In2 クラス、加速時トルク算出クラス、減速時トルク算出クラスに対してオブジェクトを生成する。また、オブジェクト間のリンクは Simulink モデルのラインを基に生成する。最後に、update メソッドを持つオブジェクトである加減速状態、トルクとトルク制御間のリンクを生成する。

図 13 のシーケンス図に示すように、トルク制御クラスが加減速状態クラスの update メソッドを呼び出すことで状態値を更新する。そして、トルククラスの update メソ

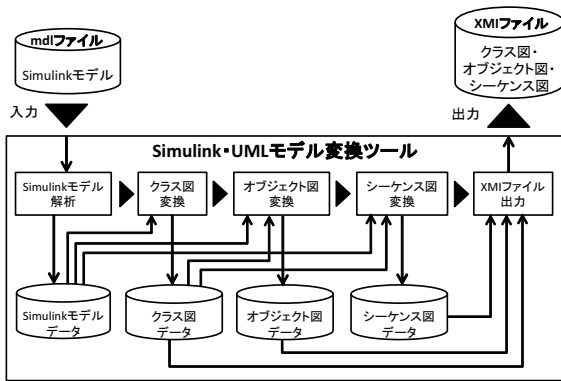


図 14 モデル変換ツールの構成

ドを呼び出し、トルククラスが加減速状態クラスの get メソッドを呼び出すことで状態値を取得する。その後、条件式に基づいて実行する処理を切り替える。状態値 (mode) がしきい値 0.5 よりも大きい場合は、加速時トルク算出クラスの update メソッドを呼び出して処理を実行させ get メソッドを呼び出すことで出力データを取得する。状態値がしきい値 0.5 よりも小さい場合は、減速時トルク算出クラスの update メソッドを呼び出して処理を実行させ get メソッドを呼び出すことで出力データを取得する。

If ブロックや SwitchCase ブロックを用いた Simulink モデルにおいても、3.1 節の考え方をを用いることで図 11～図 13 の UML モデルに変換することができる。

4. 変換ツールの実装

本研究で開発したモデル変換ツールの構成を図 14 に示す。本ツールは Simulink モデルのファイル保存形式である mdl ファイルを入力し、UML モデルの標準的なファイル保存形式である XMI ファイルを出力する。

まず、ツールは入力された mdl ファイルを解析し変換に必要なデータを抽出した Simulink モデルデータを生成する。続いて生成した Simulink データをもとに、クラス図変換を行う。クラス図変換では、Simulink モデルデータからクラス図データであるクラスや関連などを生成する。そして、Simulink モデルデータと生成したクラス図データをもとにオブジェクト図変換を行う。オブジェクト図変換では、Simulink モデルのラインからリンクをクラスからオブジェクトを生成する。その後、Simulink モデルデータとクラス図データをもとにシーケンス図変換を行う。シーケンス図変換では、Simulink モデルデータのラインデータとクラス図データのクラスからシーケンス図を生成する。最後に、クラス図データとオブジェクト図データ、シーケンス図データを XMI ファイルとして生成する。

5. 適用実験

開発したモデル変換ツールの有用性を評価するため、状態遷移や条件分岐を含むクルーズコントロール制御シス

表 1 ブロックの内訳

ブロックの種類	クルーズコントロール 制御システム	フォールトトレラント 燃料噴射システム	
		解説書	R2013
階層化前 総ブロック	21	53	31
階層化後 総ブロック	17	30	21
Inport	8	4	4
Subsystem	4	20	13
Stateflow	1	1	1
Switch	1	3	0
Outport	3	1	1
MultiportSwitch	0	1	0
SwitchCase	0	0	1
Merge	0	0	1

表 2 UML モデルの要素数

要素	クルーズコントロール 制御システム	フォールトトレラント 燃料噴射システム	
		解説書	R2013
クラス数	17	32	22
オブジェクト数	15	30	20
メッセージ数	44	288	154

テム [11] と 2 つのフォールトトレラント燃料噴射システム [12], [13] の 3 つの Simulink モデルに対して適用実験を行った。

3 つの Simulink モデルの階層化前と階層化後のブロックの内訳を表 1 に示す。3 つの Simulink モデルをそれぞれ入力し本モデル変換ツールから出力された UML モデルの各要素数を表 2 に示す。

クルーズコントロール制御システムでは Switch ブロックを用いて記述されており、選択する出力データ名が同じ場合の変換方法を適用している。また、サイバネットシステム株式会社のフォールトトレラント燃料噴射システムでは Switch ブロックと MultiportSwitch ブロックを用いて記述されており、Switch ブロックには出力データ名が同じ場合の変換方法を、MultiportSwitch ブロックには出力データ名が異なる場合の変換方法を適用している。MATLAB R2013 のサンプルモデルであるフォールトトレラント燃料噴射システムでは、MultiportSwitch ブロックを SwitchCase ブロックと Merge ブロックを用いて記述されており、選択する出力データ名が異なる場合の変換方法を適用している。

以上により、状態遷移や条件分岐を含む Simulink モデルを本モデル変換ツールに入力することで、状態遷移や条件分岐に応じて実行する処理を切り替える実行効率の良い UML モデルに変換できることを確認した。

6. 関連研究との比較

ここでは、Simulink モデルにおける対象とするブロックの範囲と UML モデルにおける対象とする図の範囲および変換方法について、第 1 章で取り挙げた 3 つの従来研究と本研究とを比較し、違いを明らかにする。

まず、Simulink モデルにおける対象とするブロックの範囲を比較する。Ramos-Hernandez らや Müller-Glaser ら、Sjöstedt らは Simulink モデルの全ブロックを対象に変換方法を提案している。

これに対し、本研究では Inport ブロック、Outport ブロック、Subsystem ブロック、Stateflow ブロック、Switch ブロック、MultiportSwitch ブロック、If ブロック、Switch-Case ブロック、Merge ブロックの 9 種類のブロックを対象としているが、構造レイヤにおいて状態遷移や条件分岐の表現に用いられるブロックを全て含んでいるため十分な実用性を持つものとする。

次に、生成対象の UML モデルの対象範囲を比較する。Ramos-Hernandez らはクラス図のみを生成対象とし、Müller-Glaser らはクラス図とコミュニケーション図を生成対象としている。また、Sjöstedt らは複合構造図とアクティビティ図を生成対象としている。

一方、本研究は構造モデルのクラス図、オブジェクト図と振る舞いモデルのシーケンス図、アクティビティ図を生成対象としておりソフトウェアの構造と振る舞いの両面を表現することができる。

最後に、変換方法を比較する。Ramos-Hernandez らや Müller-Glaser ら、Sjöstedt らはブロックやライン毎にクラスを生成しているため、Simulink モデル内に条件分岐が含まれる場合、本研究のように出力データ名に応じたクラス構成を出力することはできない。また、If ブロックや SwitchCase ブロックを含む Simulink モデルの場合、それらと Merge ブロックを別のクラスとして出力する。このため本研究で生成する UML モデルの方が再利用性に優れていると考える。

7. おわりに

状態遷移や条件分岐を含む Simulink モデルを入力し、状態遷移や条件分岐に応じて実行する処理を切り替える UML モデルを出力するモデル変換ツールを開発した。そして、複数の Simulink モデルに対して適用実験を行い、その有用性を確認した。

謝辞 本モデル変換ツールのベースを開発した神山氏、田村氏に感謝する。本研究の一部は JSPS 科研費 24500046 および 15K00084 の助成を受けたものである。

参考文献

- [1] The MathWorks : Simulink, <http://www.mathworks.com/products/simulink/>.
- [2] Sangiovanni-Vincentelli, A. and Di Natale, M.: Embedded System Design for Automotive Applications, IEEE Computer, Vol.40, No.10, pp.42-51, 2007.
- [3] Ramos-Hernandez, D.N., I.Zubizarreta, Fleming, P.J., Bennett, S. and Bass, J.M.: Towards a Control Software Design Environment Using a Meta-Modeling Technique, Proc. of 15th IFAC World Congress, 2002.
- [4] Ramos-Hernandez, D.N., Fleming, P.J. and Bass, J.M.: A Novel Object-Oriented Environment for Distributed Process Control Systems, Control Engineering Practice, vol.13, No.2, pp.213-230, 2005.
- [5] Kühl, M., Reichmann, C., Prötel, I. and Müller-Glaser, K.D.: From Object-Oriented Modeling to Code Generation for Rapid Prototyping of Electronic Systems, Proceedings of the 13th IEEE International Workshop on Rapid System Prototyping, pp.108-114, 2002.
- [6] Müller-Glaser, K.D., Frick, G., Sax, E. and Kühl, M.: Multiparadigm Modeling in Embedded Systems Design, IEEE Transactions on Control Systems Technology, Vol.12, No.2, pp.279-292, 2004.
- [7] Sjöstedt, C.-J., Shi, J., Törngren, M., Servat, D., Chen, D., Ahlsten, V. and Lönn, H.: Mapping Simulink to UML in the design of embedded systems: Investigating scenarios and structural and behavioral mapping, Proc.OMER 4 Post Workshop, 2008.
- [8] 田村雅成, 神山達哉, 添田隆弘, 兪明連, 横山孝典, Simulink モデルと UML モデルを用いた組み込み制御ソフトウェア開発のためのモデル変換環境, 情報処理学会論文誌, vol.53, No.12, pp.2660-2670, 2012.
- [9] MathWorks Automotive Advisory Board (MAAB), Control Algorithm Modeling Guidelines Using MATLAB(R), Simulink(R), and Stateflow(R), Version 3.0, 2012.
- [10] Japan MBD Automotive Advisory Board (JMAAB), Control Algorithm Modeling Guidelines Using MATLAB(R), Simulink(R), and Stateflow(R), Version 4.0, 2015.
- [11] サイバネットシステム株式会社, Simulink/Stateflow サンプルモデル解説書-クルーズコントロール制御編-, 2004.
- [12] サイバネットシステム株式会社, Simulink/Stateflow サンプルモデル解説書-フォールトトレラント燃料噴射システム編-, 2003.
- [13] The MathWorks : フォールトトレラント燃料制御システムのモデル化, <http://jp.mathworks.com/help/simulink/examples/modeling-a-fault-tolerant-fuel-control-system.html>.