

モデルベース並列化ツールを用いた モデルレベルブロック分割手法の検討

山田 亜汰朗^{1,a)} 生沼 正博^{1,2} 木村 一臣² 山本 椋太¹ 枝廣 正人¹

概要：車載システム開発では MATLAB/Simulink を用いたモデルベース開発を行っている。近年、車載システム開発は大規模化・複雑化しており、またシングルコアの性能向上は限界が近づいている。このような背景のもと、マルチコア・メニーコアの導入が注目され、自動車産業界では Simulink モデルに対し並列動作を考慮した Simulink モデルに分割することが要求されている。そこで本稿では、Simulink モデルに対し動作を変えないように自動でモデル分割を行う ModelSplitter ツールを提案する。本提案により、Simulink モデルを並列動作可能な複数の Simulink モデルに分割することが可能となった。

1. はじめに

近年、車載システム開発は大規模化・複雑化している。車載システム開発では MATLAB/Simulink[1] を用いたモデルベース開発 (MBD) が広く用いられている。一方、シングルコアの性能向上の限界により、マルチコア・メニーコアによる研究・開発が進められている。

自動車産業界では OEM (自動車メーカー) と Supplier (自動車部品メーカー) の協業により開発が行われている。OEM が Simulink モデルを設計し、Supplier がリリースされた Simulink モデルをターゲットへ実装する。マルチコア・メニーコア開発において、OEM では Simulink モデルをマルチコア上に配置できるかどうか判断することが困難であり、Supplier ではリリースされた Simulink モデルが並列動作未考慮による性能未達を判断することが困難である。このため、自動車産業界では並列動作を考慮した Simulink モデルの分割が要求されている [3]。そこで本稿では、Simulink モデルに対し動作を変えないように自動でモデル分割を行う ModelSplitter ツールを提案する。

我々の研究グループでは、MBD による並列化システム開発支援ツールであるモデルベース並列化 (MBP) ツールの開発を進めてきた [3]。本稿で提案する、MBP ツールを用いた並列化設計フローを図 1 に示す。図 1 (1) では Simulink モデルを入力として、ブロックレベル構造情報 (BLXML) を生成する MBP ツールを使用する。図 1 (2) では BLXML を入力として、ブロックの情報およびブロック間の構造情報を用いてコア割当て情報を BLXML に付

加する MBP ツールを使用する。図 1 (3) では Simulink モデルとコア割当て情報付 BLXML を入力として、並列動作可能な分割 Simulink モデルを生成する ModelSplitter ツールを使用する。ModelSplitter は本稿で提案するツールである。同一コアに割当てられたブロックをいくつかずつ Atomic Subsystem 化し、モデルリファレンスに変換して、複数の Simulink モデルを生成する。分割によって動作を変えないように、信号線で繋がったブロック同士で分割を判定する。Simulink モデル分割が完了すると分割前の Simulink モデルと等価性検証を行う。

2. ModelSplitter モデルレベル分割フロー

ModelSplitter の詳細を図 2 で示す。

図 2 (1) について述べる。図 1 で示したコア割当て付 BLXML を入力としてブロック情報テーブルを生成する。ブロック情報テーブルの例を図 3 に示す。ブロック情報テーブルはモデル分割に必要な情報を抽出し、新たに分割に必要な情報を追加するために使用する。BLXML に含まれる各ブロックに対して 1 つの要素を構成する。その構成

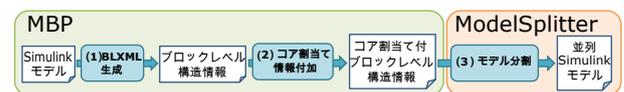


図 1 MBP を用いた並列化設計フロー

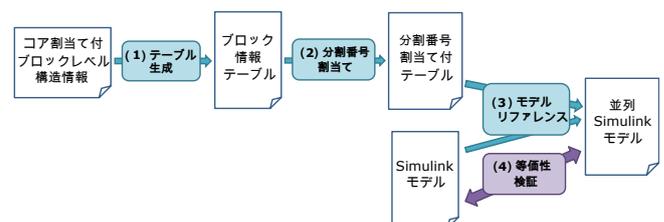


図 2 ModelSplitter を用いたモデルレベルブロック分割フロー

¹ 名古屋大学
Nagoya University, Nagoya, Aichi 464-8601, Japan
² ガイオ・テクノロジー株式会社
GAIO TECHNOLOGY CO., Ltd, Shinagawa, Tokyo 140-0002, Japan
a) ataro_y@ertl.jp

ブロック	コア割当て番号	入力ブロック	出力ブロック	分割フラグ	分割番号
ブロックタイプ	type	peinfo	input	output	flag refnum
test1_Add	'Sum'	0	{'test1_Constant', 'test1_Gain'}	{}	1 0
test1_Constant	'Constant'	1	{}	{'test1_Add'}	1 0
test1_Constant1	'Constant'	1	{}	{'test1_Gain'}	1 1
test1_Gain	'Gain'	0	{'test1_Constant1'}	{'test1_Add', 'test1_Gain1'}	1 0
test1_Gain1	'Gain'	0	{'test1_Gain'}	{'test1_Display'}	1 0
test1_Display	'Display'	1	{'test1_Gain1'}	{}	1 2

図 3 ブロック情報テーブルの例

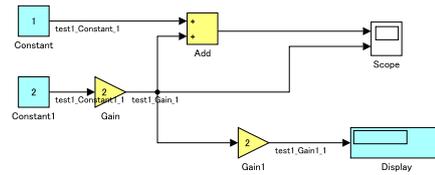


図 4 モデル分割前

はブロック、ブロックタイプ、コア割当て番号、入力ブロック、出力ブロック、分割フラグおよび分割番号である。分割フラグは、各ブロックに対し分割の対象にするかどうか判定するフラグである。このフラグは図 2 (3) のフェーズで同階層に存在するブロックのみでサブシステム化を行うため使用する。特にサブシステムを含むモデル分割では、サブシステムブロックとその中身のブロックでは階層が異なり、どちらかのみを分割の対象とするため必要である。分割番号は図 2 (2) のフェーズで割当て、図 2 (3) のフェーズで使用する。

図 2 (2) について述べる。各ブロックに対してコア割当て番号ごとに分割番号を割当て、ブロック情報テーブル内の値を更新する。分割番号割当ては以下の流れで行う。

- コア割当て番号ごとに初期値 0 として分割番号を用意し (b) に移動。
- ブロック情報テーブルの上から順に分割番号未割当てのブロックを選択 (基準ブロックと呼ぶ) し (c) に移動。分割番号未割当てのブロックが存在しない場合、終了。
- 基準ブロックのコア割当て番号を取得 (基準コア番号と呼ぶ) し、基準コア番号の分割番号を基準ブロックに割当て (d) に移動。
- 選択したブロックに入力ブロックまたは出力ブロックが存在すればそのブロックを選択し (e) に移動。存在しなければ (f) に移動。
- 選択したブロックのコア割当て番号が基準コア番号と一致すれば基準ブロックの分割番号を割当て (d) に移動。一致しなければ (d) で移動する前のブロックに戻り (d) に移動。
- 基準ブロックの分割番号をインクリメントし (b) に移動。

図 2 (3) について述べる。コア割当て番号ごとに割当てた分割番号に従ってサブシステム化を行う。サブシステム化後、Simulink のモデルリファレンスを使用するために Atomic サブシステム化を行う。また、モデル分割前後の等価性検証を行うためにサブシステムの出力信号線のログをオンにする。作成した Atomic サブシステムをモデルリファレンスに変換することで複数の Simulink モデルに分割する。

図 2 (4) について述べる。シミュレーションデータインスペクター (sdi) [2] を用いて等価性検証を行う。等価性検証では分割前と分割後の Simulink モデルに対し、同じ入力データを使用してシミュレーションを行う。出力されたデータが分割前と分割後で一致していることを確認する。

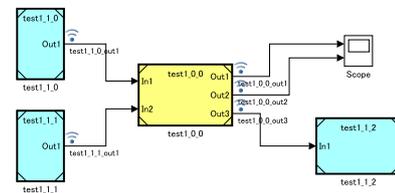


図 5 モデル分割後

その結果、分割が正しく行われたかどうかを確認できる。

3. ModelSplitter 動作確認

検証対象 Simulink モデルを図 4 に示す。本検証ではコア割当てを手動で行っており、2 コア割当てを色分けで表現している。ただし、Scope ブロックは MBP ツールでコア割当て対象外である。

図 4 のモデルを ModelSplitter に適用した分割後のモデルを図 5 に示す。同一コアに割当てられたブロックを意図したとおりに複数の Simulink モデルに分割することができた。また、sdi を用いた等価性検証では分割前後でシミュレーション結果が変わらないことを確認できた。よって、動作を変えずにモデル分割できたことが確認できた。

4. まとめと今後の課題

本研究では、モデルベース並列化ツールを用いたモデルレベルブロック分割手法を検討した。Simulink モデルを並列動作可能な複数の Simulink モデルに分割することが可能となった。

現状では、いくつかのテストモデルに対して検証を行った。今後はより様々なモデルに対して同検証を行い、アルゴリズムの強化を試みる予定である。

参考文献

- MathWorks: Simulink, available from <<https://jp.mathworks.com/products/simulink.html>> (2020 年 11 月 13 日参照) .
- MathWorks: シミュレーションデータインスペクター, available from <<https://jp.mathworks.com/help/simulink/slref/simulationdatainspector.html>> (2020 年 11 月 13 日参照) .
- 生沼正博, 山本椋太, 竹内成樹, 権藤正樹, 本田晋也, 近藤真己, 枝廣正人: モデルベース並列化ツールを用いたマルチコアシステム開発フローの提案, DA シンポジウム 2020 論文集, Vol. 2020, pp. 73-80 (2020).