一般社団法人 電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
IEICE Technical Report

Simulink モデルにおけるグラフに基づく非完全一致モデルクローン検出

鷲崎　弘宜† 　村上　真一† 　深澤　良彰†

† 早稲田大学基幹理工学部情報理工学科

早稲田大学グローバルソフトウェアエンジニアリング研究所

〒169-8555 東京都新宿区大久保 3-4-1

E-mail: †washizaki@waseda.jp　s.murakami-620@fuji.waseda.jp　fukazawa@waseda.jp

あらまし　Simulink モデルは，ブロック線図としてプログラムを表現したモデルであり，制御系を中心として組込みソフトウェア開発において使われつつある．大規模なモデルや，同一ドメインで複数のモデルを扱う場合，クローンと呼ばれる重複箇所の存在がモデルの保守性を低下させる可能性がある．この問題解決に向けて，完全に一致するクローンを検出する手法が提案されているが，コピー＆ペースト後に部分的に変更されたような非完全一致ながら類似性の高いクローン（ギャップを含むクローン）を十分に検出できなかった．そこで我々は，完全一致のクローンを検出する既存の手法と，多頻度グラフ検出アルゴリズムを組み合わせることで，非完全一致のモデルクローンを Simulink モデル群から効率よく検出する手法を提案する．提案手法を検出ツールとして実装し，複数の具体的な制御モデルに適用した結果，非完全一致のクローンを検出可能なことを確認した．提案手法により検出した結果を記録し管理することで，効率的な保守を実現することが期待できる．

キーワード　Simulink，モデリング，制御モデル，クローン検出

Graph-based Detection of Imperfect-Matching Clones in Simulink Models

Hironori Washizaki †　Shinichi Murakami ‡　and　Yoshiaki Fukazawa ‡

† Waseda University, Dept Computer Science and Engineering, Global Software Engineering Laboratory

3-4-1, Okubo, Shinjuku-ku, Tokyo, 169-8555 Japan

E-mail: †washizaki@waseda.jp　s.murakami-620@fuji.waseda.jp　fukazawa@waseda.jp

Abstract　Simulink models represent program as block diagrams for usually control system design. In large scale models or a large number of models, duplicated parts called “clones” could reduces the maintainability of models. To overcome such issue, there are existing researches on detecting perfect-matching clones; however these researches are not adequate for detecting imperfect-matching clones made by some partial modifications after copied and pasted. We propose a technique for detecting imperfect-matching clones in Simulink models efficiently by combining the existing perfect-matching clone detection technique and a fast apriori-based graph mining algorithm. We implemented the proposed technique as a detection tool and confirmed that the tool can detect imperfect-matching tools by applying it to several control system models. It is expected that models could be easily maintained by detecting such clones by the technique, and recording and managing them.

Keyword　Simulink，Modeling，Control models，Clone detection

1. はじめに

自動車業界の企業をはじめとして，組み込みソフトウェア開発の現場ではモデルベース開発(Model Based Development:MBD)が広く行われている．MBD においてモデルを作成，検証するツールとして MATLAB/Simulink[1] が広く使われている．MATLAB/Simulink によって生成された Simulink モデルと呼ばれる Stateflow 図は，MATLAB/Simulink 上で動作のシミュレーションができるブロック線図としてプログラムを表現したモデルである．Simulink モデル作成の過程においては，ソースコード作成の過程においてと同様，他の箇所からのコピー＆ペーストがしばしば行われる．これによって発生した，ソフトウェア内における同一のブロック群をモデルクローンと呼ぶ．モデルクローンは同一であるにもかかわらず一元管理されておらず，統一して行われるべき修正が困難であるなど，保守性に大きな問題を引き起こしている[2][3].

この問題を解決するため，Simulink モデルにおけるモデルクローンを検出する研究が行われており，その成果は ConQAT[4][5]のようなツールの機能として一般に提供されているものもある．しかし，従来の研究

では完全一致のモデルクローン検出[5][6][7]が主であり，コピー&ペースト後に変更が加えられたために完全一致のモデルクローンではなくなったものの，依然として類似性が高く，統一的に管理されるべき大きな構造の類似した非完全一致のモデルクローン（ギャップクローン）を検出する手法は十分でなかった．

そこで我々は，完全一致のモデルクローンの検出結果をグラフ構造で整理した上で，グラフマイニングのアルゴリズムである AGM アルゴリズム[8]を利用し，Simulink モデルから非完全一致のモデルクローンを検出する手法を提案する．

2. Simulink モデルクローン問題

MATLAB/Simulink[1]上では，数値計算プログラムを機能単位でブロックとしてグラフィカルに表示し，GUI を使ってシミュレーションが実行できるモデルとして組み立てていくことができる．こうして MATLAB/Simulink 上で作られたブロック線図を Simulink モデルと呼ぶ．ブロック群は任意の個数で新たな 1 ブロックとしてまとめることができる．これを Subsystem と呼ぶ．以降において Simulink モデルにおけるクローンについて述べる．

2.1. 完全一致のモデルクローン

モデルを作成する過程において，処理の一部として必要となる他箇所と同一の構造を，他箇所からのコピー&ペーストによって作成することがしばしばある．このソフトウェア内における同一の構造を持ったブロック群を，モデルクローンと呼ぶ．以下，次節で説明する非完全一致のモデルクローンと区別するため，一般にモデルクローンと呼ばれているブロック群の構造が完全に一致しているモデルクローンを，完全一致のモデルクローンと呼ぶ．

Simulink モデルにおける完全一致のモデルクローンの例を，図 1 に示す． 完全一致のモデルクローンは，ブロック群の構造が完全に一致しているにもかかわらず一元管理されていない．その結果として，部品化されていないことによる開発効率の低下や，統一して行われるべき修正が困難なことによる保守性の低下といった問題を引き起こしている．

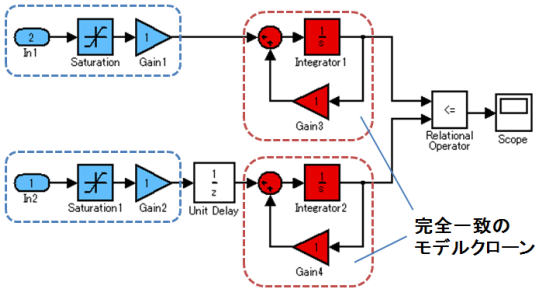

図 1: 完全一致のモデルクローン

Simulink モデルを対象とした完全一致のモデルクローン検出に関する研究の提案手法を，機能の一つとして実装しているツールが複数存在する．ConQAT は，それらの中でも非常に有名かつ多機能なツールである．ConQAT は，Simulink モデルのモデルクローンの検出機能以外にも，Java や C，C#をはじめとした多くの言語を対象に，ソースコードのクローン検出機能や，モデルやソースコードの品質を様々なメトリクスで評価する機能を備えており，ソフトウェアの総合分析ツールとして知られている．

我々は，非完全一致のモデルクローンを検出する提案手法の中で，完全一致のモデルクローンの検出結果を用いるため，その工程で ConQAT を利用する．ConQAT が完全一致のモデルクローンを検出する処理手順を，以下で簡単に説明する．

1. 構造解析: Simulink モデルのモデルファイルを独自の解析器にかけ，ブロック線図の構造を簡略化したグラフに変換する．その際，あくまで構造の同一性にのみ注目するため，ブロックごとに与えられるパラメータなどの情報は無視される．
2. 検出処理: ノード数 k=1 から開始し，複数箇所に存在が確認される構造を検出していく．得られた結果を入力とし，k→k+1 として操作を繰り返すことで，次第に大きなノード数のモデルクローンが検出されるようになる． 処理の開始時にユーザが指定した，検出対象とするモデルクローンの最大ノード数の情報に従って処理を終了する．
3. 出力: モデルクローンとして検出された構造のサイズ，構造，検出箇所数と，検出箇所を Simulink モデルのモデルファイルを着色することで直接示す MATLAB/Simulink のコマンドファイルを出力する．

2.2. 非完全一致のモデルクローン

モデルクローンの中には，コピー&ペースト後に変更が加えられたために完全一致のモデルクローンではなくなったものの，依然として類似性が高く，統一的に管理されるべきモデルクローンが存在する．これを非完全一致のモデルクローンと呼ぶ．非完全一致のモデルクローンの例を，図 2 に示す． 非完全一致のモデルクローンは，その構造が完全一致ではないため，従来の多くの研究によるモデルクローン検出手法では検出することができない．

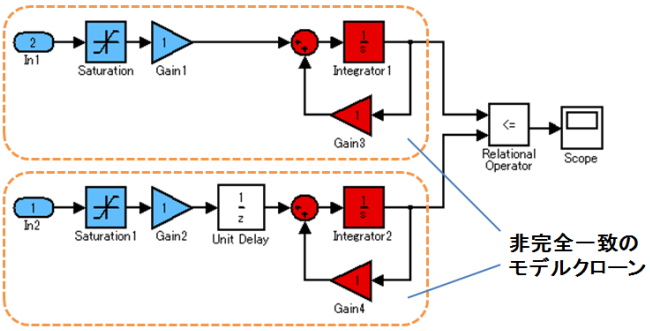

図 2: 非完全一致のモデルクローン

3. 非完全一致クローン検出

既存の研究では非完全一致のモデルクローンを検出できない問題を解決するため，我々は完全一致のモデルクローンの検出結果と，グラフマイニングの分野で部分グラフ同型問題を解く代表的なアルゴリズムとして知られる AGM アルゴリズムを用いて，非完全一致のモデルクローンを検出する手法を提案する．

提案手法は，次の 4 つのステップ①～④で構成されている．

① 完全一致のモデルクローン検出: 対象とする Simulink モデルから完全一致のモデルクローンを検出する．このステップでは既存のツール（ConQAT）を利用する．

② 完全一致のモデルクローンの位置関係グラフ化: 対象とする Simulink モデルに含まれる Subsystem ごとに，完全一致のモデルクローンの位置関係をグラフ化する．この際，完全一致のモデルクローンとして検出された構造以外のすべてのブロックを無視して考え，完全一致のモデルクローンのみの位置関係をグラフ化する．

③ 非完全一致のモデルクローンパターン検出: 前ステップで Subsystem ごとに作成したグラフの集合を対象に, AGM アルゴリズムを用いて部分グラフ同型判定を行い，グラフの集合から多頻度グラフパターンを検出する．ここで検出された多頻度グラフパターンが，非完全一致のモデルクローンの構造の共通部分を表すパターンとなる．

④ 非完全一致のモデルクローン出力: 検出された多頻度グラフパターンから，実際にその構造が現れる部分を特定し，非完全一致のモデルクローンとなるブロック群を抽出して，その位置情報と構造を出力する．

提案手法の全体像を図 3 に示す．また，図 4，図 5 に示す Simulink モデルを入力とした場合を例に，各ステップでどのような処理が行われるかを追って，提案手法の詳細を以降の節にて説明する．

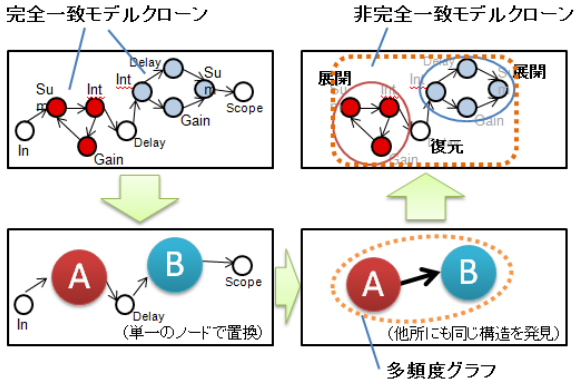

図 3: 提案手法の全体像

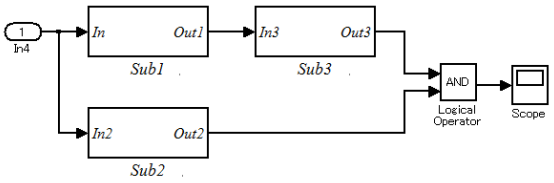

図 4: 入力 Simulink モデルの全体図

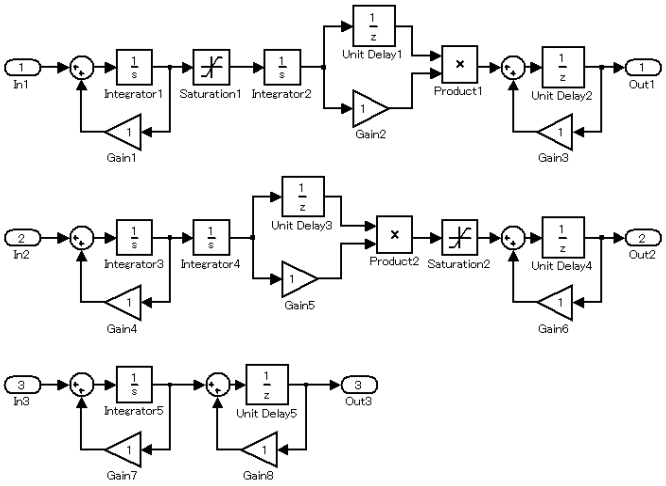

図 5: 入力モデルの各 Subsystem の詳細

3.1. ①完全一致のモデルクローン検出

単一，もしくは複数の Simulink モデルファイル（MDL ファイル）を入力とし，完全一致のモデルクローンとなっているブロック群の構造を検出する．この工程では，前述の既存ツールである ConQAT を利用する．

実際に図 4 で示した Simulink モデルを入力とした場合に ConQAT によって検出される 3 種類 8 箇所の完全一致のモデルクローンの構造と位置を示した結果を図 6 に示す．

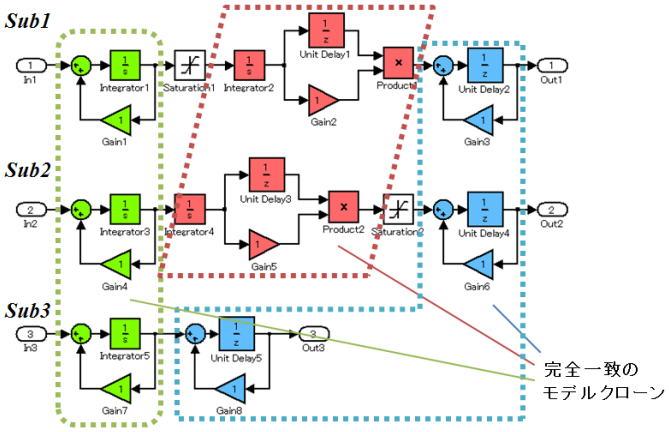

図 6: 完全一致のモデルクローン

3.2. ②完全一致のモデルクローンの位置関係グラフ化

Subsystem ごとに完全一致のモデルクローンの位置関係をグラフ化する．この工程は，次の 3 つのステップ i～iii で構成される．

i. 単一ノードによる置き換え 完全一致のモデルクローンを構成しているブロック群を，モデルクローンの種類ごとに単一のノードに置き換える．

ii. 完全一致のモデルクローン以外のブロックの除去 完全一致のモデルクローンのみの位置関係をグラフ化するため，完全位置のモデルクローンを構成しているブロック群以外のブロックを一時的に除去する．

iii. 隣接行列による表現 ステップ 2 によって得られたグラフを，隣接行列によって表現する．

以下，実際に図 4 の Simulink モデルからどのようなグラフ構造データが得られるかについて，各ステップごとに述べる．

3.2.1. ②-i 単一ノードによる置き換え

図 5 の Simulink モデルを例に，完全一致のモデルクローンの位置関係をグラフ化する手順を説明する．まず，3.1 節で得られた完全一致のモデルクローンを構成しているブロック群を，モデルクローンの種類ごとに単一のノードに置き換える．この際，置き換える単一のノードに，それぞれ固有の ID を付しておく．今回は，各ブロック群を以下のように命名する．

A：“Sum, Integrator, Gain”

B：“Integrator, Delay, Gain, Sum”

C：“Sum, Delay, Gain”

この命名に従い，図 5 の Simulink モデルの該当する部分を単一のノードに置き換えたものを，図 7 に示す．

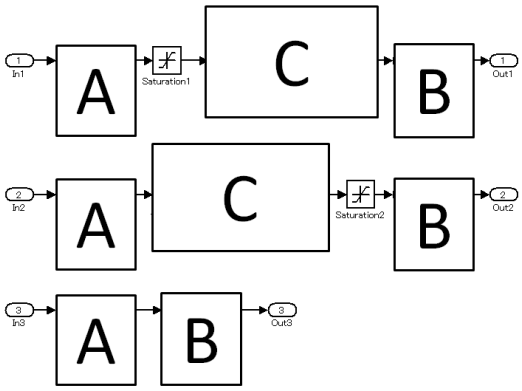

図 7: 単一のノードによる置き換え

3.2.2. ②-ii 完全一致クローンの単一ノードによる置き換え

非完全一致のモデルクローンを検出する準備として，完全一致のモデルクローンのみの位置関係をグラフ化しておく必要がある．そのため，完全位置のモデルクローンを構成しているブロック群以外のブロックを一時的に除去する．

図 7 に示した Simulink モデルから，完全一致のモデルクローンを構成しているブロック群以外のブロックを除去した Simulink モデルのグラフ表現を図 8 に示す．

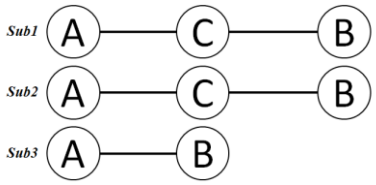

図 8: 完全一致クローン以外のブロックの除去

3.2.3. ②-iii 隣接行列による表現

前ステップで得られたグラフを入力としてグラフマイニングを行うために，Simulink モデルを，Subsystem ごとに隣接行列による表現に直す．

図 8 の場合はまず，単一のノードとして置き換えられた頂点ラベル A，B，C に対し，1，2，3 を割り当て，辺ラベルは 1 を割り当てる．Sub1，Sub2，Sub3 を表した隣接行列を以下に示す．

$$Sub1 = \begin{array}{c} \\ C \\ B \\ A \end{array} \begin{array}{c} \begin{array}{ccc} C & B & A \end{array} \\ \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix} \end{array} \quad Sub2 = \begin{array}{c} \\ C \\ B \\ A \end{array} \begin{array}{c} \begin{array}{ccc} C & B & A \end{array} \\ \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix} \end{array}$$

$$Sub3 = \begin{array}{c} \\ C \\ B \end{array} \begin{array}{c} \begin{array}{cc} C & B \end{array} \\ \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \end{array}$$

3.3. ③非完全一致のモデルクローンパターン検出

3.2 節で Subsystem ごとに作成した完全一致のモデ

ルクローンの位置関係を表したグラフの集合を対象に，AGM アルゴリズムを用いて部分グラフ同型判定を行い，グラフの集合から多頻度グラフパターンを検出する．ここで検出された多頻度グラフパターンが，非完全一致のモデルクローンの構造の共通部分を表すパターンとなる．この工程は，次の 5 つのステップで構成される．

i. 隣接行列結合: サイズ k の多頻度グラフパターンを組み合わせて，サイズ k+1 の多頻度グラフの候補を生成する．結合された隣接行列は，新たに結合されたノード間のリンクの有無によって，複数の正規形のグラフとして生成される．

ii. 部分グラフチェック: 結合された隣接行列が多頻度グラフであるためには，そのすべての誘導部分グラフが多頻度グラフでなくてはならない．結合された隣接行列の誘導部分グラフが多頻度グラフであるかどうかをチェックする．

iii. 正準化: 結合された隣接行列の正規形を正準化する．

iv. 頻度計算: 多頻度グラフパターンの候補の支持度を求める．

v. 非完全一致のモデルクローン出力: 得られた多頻度グラフパターンから，具体的に非完全一致のモデルクローンとなっているブロック群を特定し，出力する．

例として，実際に 3.2.3 節で得られた隣接行列をもとに，ステップ i～iv を実施すると，最低支持度を 60%と指定したとき，以下の 6 つの隣接行列が多頻度グラフとして出力される．

$$\begin{array}{c} \\ A \\ B \\ C \end{array}\begin{array}{c} \begin{array}{c} A \end{array} \\ \begin{pmatrix} 0 \end{pmatrix} \\ \begin{pmatrix} 0 \end{pmatrix} \\ \begin{pmatrix} 0 \end{pmatrix} \end{array}\quad \begin{array}{c} \\ B \\ C \\ A \end{array}\begin{array}{c} \begin{array}{c} B \end{array} \\ \begin{pmatrix} 0 \end{pmatrix} \\ \begin{pmatrix} 0 \end{pmatrix} \\ \begin{pmatrix} 0 \end{pmatrix} \end{array}\quad \begin{array}{c} \\ C \\ A \\ B \end{array}\begin{array}{c} \begin{array}{c} C \end{array} \\ \begin{pmatrix} 0 \end{pmatrix} \\ \begin{pmatrix} 0 \end{pmatrix} \\ \begin{pmatrix} 0 \end{pmatrix} \end{array}$$

$$\begin{array}{c} \\ C \\ A \end{array}\begin{array}{c} \begin{array}{cc} C & A \end{array} \\ \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \end{array}\quad \begin{array}{c} \\ C \\ B \end{array}\begin{array}{c} \begin{array}{cc} C & B \end{array} \\ \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \end{array}\quad \begin{array}{c} \\ C \\ B \\ A \end{array}\begin{array}{c} \begin{array}{ccc} C & B & A \end{array} \\ \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix} \end{array}$$

さらにここで，検出する多頻度グラフの最低サイズを 3 としてフィルタリングすることによって，以下の隣接行列のみに絞ることができる．

$$\begin{array}{c} \\ C \\ B \\ A \end{array}\begin{array}{c} \begin{array}{ccc} C & B & A \end{array} \\ \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix} \end{array}$$

最終的に得られた隣接行列を元に，一時的に除去していた完全一致のモデルクローンを構成するブロック群以外のブロックを復元し，非完全一致のモデルクローンとして出力する．

上述の隣接行列は，図 8 で Sub1, Sub2 に存在する“A-C-B”のパターンを意味している．そこで，Sub1, Sub2 内の“A-C-B”のパターン部からブロックを復元し，非完全一致のモデルクローンとして出力する．非完全一致のモデルクローンとして出力される構造を図 9 に示す．

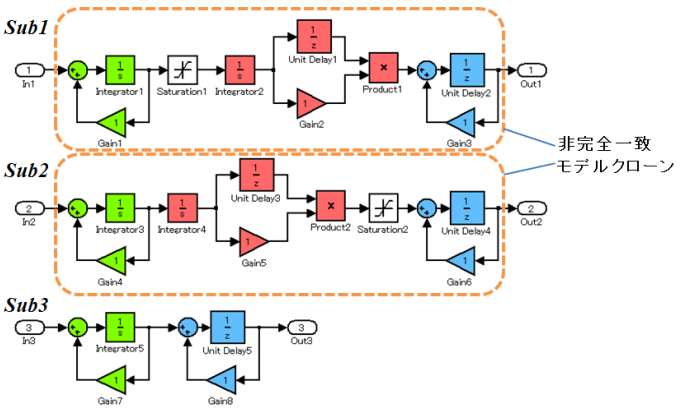

図 9: 非完全一致のモデルクローン

4. ツール実装と適用実験

我々は提案手法をツールとして実装した．ツールは，Simulink モデルを入力として，完全一致のモデルクローンの検出に ConQAT を用いる．また，AGM アルゴリズムによる多頻度グラフの検出には，汎用のデータマイニングツールである MUSASHI[9]を用いた．

提案手法の有効性を検証するため，総ブロック数 59，ブロック種 12，Subsystem 数 5 からなる一定規模のモデルを作成して適用実験を行った．入力モデルの全体を図 10 に，各 Subsystem の内容を詳細図として図 11 に示す．このモデルからは 4 種類 15 箇所の完全一致のモデルクローンが検出された．この結果を元に，完全一致のモデルクローンの位置関係を表したグラフを生成し，多頻度グラフを検出して，作成時に想定した全ての 2 種類 4 箇所の非完全一致のモデルクローンを検出することに成功した．

完全一致および非完全一致のモデルクローンの構造と位置を合わせて図 11 に示す．また結果をまとめた表を図 12 に示す．この結果より，一定規模の Simulink モデルにおいて，提案手法により非完全一致モデルクローンを検出可能なことを確認した．このような一定規模のモデルにおいて類似かつ大きな構造を自動的に特定することにより，全体の構成を理解しやすくすると同時に，以降の保守における修正必要個所の特定や修正・拡張方法の識別がしやすくなると考えられる．

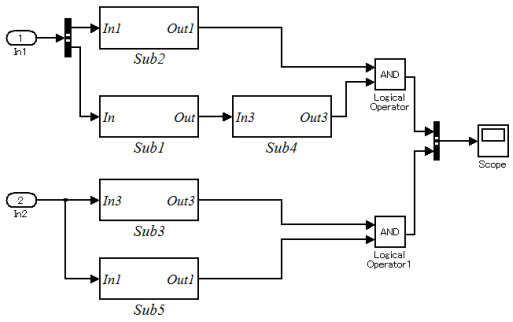

図 10: 入力 Simulink モデルの全体図

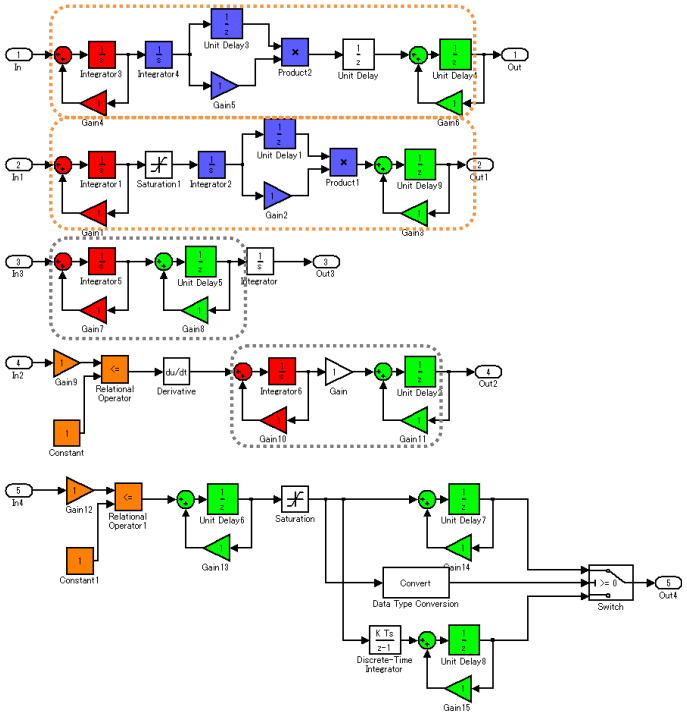

図 11: モデルクローンの構造と位置（濃淡: 完全一致クローン，点線囲い: 非完全一致クローン）

Simulinkモデル構成ブロック		完全一致モデルクローン		非完全一致モデルクローン	
種類数	総数	種類数	総数	種類数	総数
12	59	4	15	2	4

図 12: モデルクローンの検出結果

5. 関連研究

Bakr Al-Batran らは，ブロックの配列や使用ブロックの種類，個数，結びつきなどといった構造には違いがあるものの，実現している機能，意味という視点では同一のものを意味的モデルクローンとして定義し，これを検出する手法を提案している[11]．また Nam H.Pham らは，完全一致のモデルクローンと，非完全一致のモデルクローンを検出する独自のグラフマイニングアルゴリズムを提案している[12]．ただし，Nam H.Pham らの定義する非完全一致のモデルクローンは，あくまでブロックの配置やパラメータの違いをもって非完全一致としており，本論文で定義した非完全一致のモデルクローンとは異なる．

これらの意味的モデルクローンや Nam H. Pham らの非完全一致モデルクローンは，基本的に完全一致モデルクローンと同規模の小さなものに限られる．一方，本論文の提案手法は完全一致モデルクローンを組み合わせたより大きな類似構造としての非完全一致モデルクローンを検出する点が異なる．今後，各手法を併用あるいは統合する可能性を検討したい．

6. おわりに

我々は，MATLAB/Simulink の Simulink モデルを対象として，従来モデルクローンとして扱われることのなかった大きな粒度で構造の類似する非完全一致のモデルクローンを検出する手法を提案した．適用実験として，用意した一定規模の Simulink モデルに提案手法を適用し，非完全一致のモデルクローンが検出されることを確認した．今後は，大規模な実システムを対象に適用実験を行うことで提案手法の実用性を検証する予定である．

文　献

[1] MATLAB/Simulink, http://www.mathworks.de/products/simulink/

[2] Bakr Al-Batran, Bernhard Schatz, Benjamin Hummel, “Semantic clone detection for model-based development of embedded systems,” In Proceedings of the 14th international conference on Model driven engineering languages and systems. (MoDELS 2011), pp.258-272, 2011.

[3] Nam H. Pham, Hoan Anh Nguyen, Tung Thanh Nguyen, Jafar M. Al-Kofahi, Tien N. Nguyen, "Complete and Accurate Clone Detection in Graph-based Models", In 31th International Conference on Software Engineering (ICSE 2009), pp 276-286, 2009

[4] ConQAT, http://www.conqat.org/

[5] Juergens, E.; Deissenboeck, F.; Hummel, B. ,CloneDetective - A workbench for clone detection research” In 31th International Conference on Software Engineering (ICSE 2009), pp.603-606, 2009.

[6] Deissenboeck, F., Hummel, B., Jurgens, E., Pfahler, M., Schatz, B.: “Model clone detection in practice.” In: International Workshop on Software Clones. IWSC ’10, pp. 57-64, 2010.

[7] Deissenboeck, F., Hummel, B., Jurgens, E., Schatz, B., Wagner, S., Girard, J.F., Teuchert, S.: “Clone Detection in Automotive Model-Based Development.” In: International Conference on Software Engineering , 2008.

[8] Inokuchi, A., Washio, T., and Motoda, H, “An Apriori-Based Algorithm for Mining Frequent Substructures from Graph Data”, In Proc. of the 4th European Conference on Principles and Practice of Knowledge Discovery in Databases(PKDD 2000), pp.13-23, 2000.

[9] MUSASHI, http://musashi.sourceforge.jp/