

Simulink モデルにおける非完全一致のクローンの検出

鷲崎 弘宜†, 村上 真一†, 深澤 良彰†

アブストラクト 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み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。

Extracting Imperfect-Matching Clones from Simulink Models

Hironori Washizaki†, Shinichi Murakami†, Yoshiaki Fukazawa†

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 reduce 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 design models. It is expected that models could be easily maintained by detecting such clones by the technique, and recording and managing them.

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 とは、従来のソフトウェア開発のようなソースコードによる開発とは異なり、開発の初期段階で実現すべき機能を設計図・モデルで作成し、開発の上流工程から下流工程において、これを検証しながら開発プロセスを進めていく開発手法のことで、開発効率や保守性、ソフトウェア資産の再利用性の向上といった利点がある。

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り、その性質は「動く仕様書」とも言われ高い評価を受けている。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、保守性に問題を引き起こすことが指摘されている[2][3]。

この問題を解決するため、Simulink モデルにおける

†早稲田大学基幹理工学部情報理工学科
グローバルソフトウェアエンジニアリング研究所
Waseda University, Dept. Computer Science and Engineering,
Global Software Engineering Laboratory.

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][6]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][7][8][9]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非完全一致のモデルクローン(ギャップを含むモデルクローン)を大きな粒度で検出する手法は十分でなかった。

そこで我々は、非完全一致のモデルクローンに焦点を当てて、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[10]を利用し、Simulink モデルから、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手法を提案する。

以降において 2 節でモデルクローンの問題を説明し、3 節で非完全一致モデルクローン検出手法を提案する、4 節ではその実装と適用実験結果を説明し、5 節で関連研究と比較したうえ、6 節で結論と展望を述べる。

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。処理のプログラミングに必要な代表的な計算式の多くがブロックライブラリとして提供されており、通信、制御、動画像処理、静止画像処理を含め、時間依存システムの設計、シミュレーション、実装、テストなどを可能としている。また、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を説明する。

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、2.2 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、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。

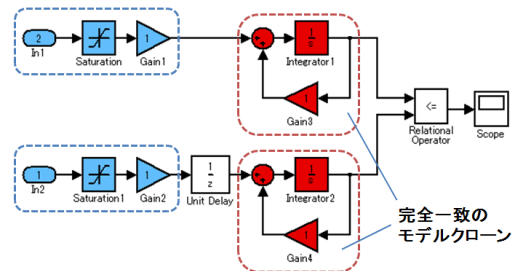


図 1: 完全一致のモデルクローン

Simulink モデルを対象として完全一致のモデルクローンを検出するツールが複数存在し、その中で ConQAT が代表的である。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、複数箇所が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、 $k \rightarrow k+1$ 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して示す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。図 2 における非完全一致モデルクローンは、複数の完全一致モデルクローンを複

合させた比較的大きな類似構造を示し、その把握により全体の理解を容易化すると同時に、全体の将来の保守の容易化が期待できる。このような大きな類似構造としての非完全一致モデルクローンは、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。

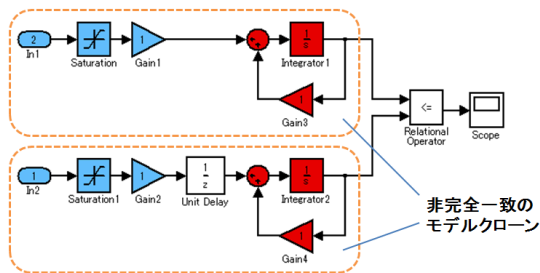


図 2: 非完全一致のモデルクローン

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 AGM アルゴリズム (Apriori アルゴリズム[11]) をグラフ構造データに拡張したものを用いて、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、共通の部分グラフをしらみつぶしに探して多頻度グラフを検出することを試みると、計算量が莫大なものとなる。そこで我々は、計算途中で比較対象を絞り込むことにより計算量を抑える AGM アルゴリズムを適用するこ

ととした。

- 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。

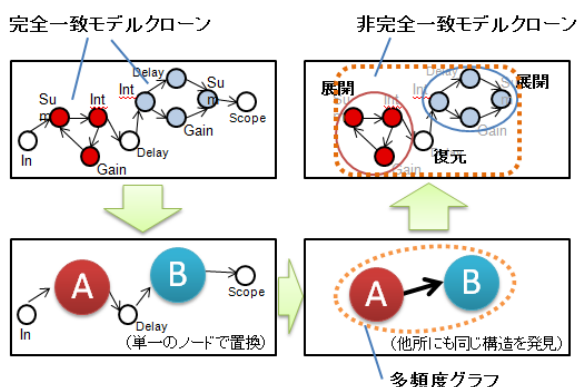


図 3: 提案手法の全体像

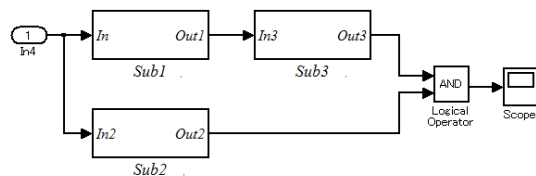


図 4: 入力 Simulink モデルの全体図

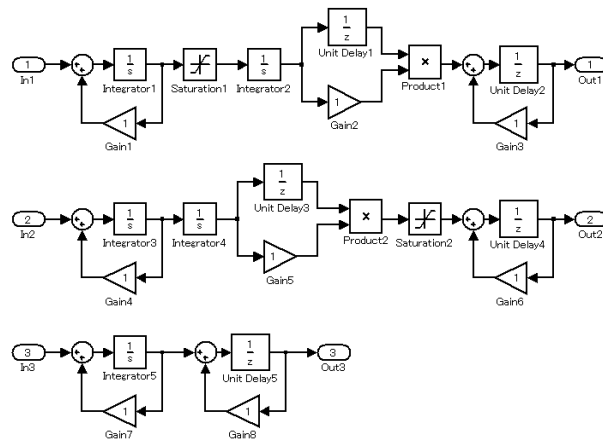


図 5: 入力モデルの各 Subsystem の詳細

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時の、ConQAT 上の結果表示を図 6 に、同時に出力される MATLAB/Simulink 上で完全一致のモデルクローンとなっている部分の位置を確認するためのコマンドファイルを利用して、検出された 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 7 に示す。

図 6 に示す結果から、4 つのブロックから成り Element ID “Clone 0” とされる完全一致のモデルクローンがモデル内に 2 箇所存在すること、3 つのブロックから成り Element ID “Clone 1” とされる完全一致のモデルクローンがモデル内に 3 箇所存在すること、3 つのブロックから成り Element ID “Clone 2” とされる完全一致のモデルクローンがモデル内に 3 箇所存在することがわかる。またコマンドファイルを利用することで図 7 に示すように、3 種類の完全一致のモデルクローンが、“Sum, Integrator, Gain” の 3 つのブロックによる構造と、“Integrator, Delay, Gain, Sum” の 4 つのブロック、そして、“Sum, Delay, Gain” の 3 つのブロックによる構造であることがわかる。

Model Clones (Clones)									
Model Clones									
Element	volume	weight sum	occurrences	size	weight	models	color-file	Layouted Clone	
Clone 0	8	8	2	4	4	clone_test	m-file	Clone 0	Clone 1
Clone 1	9	9	3	3	3	clone_test	m-file	Clone 0	Clone 1
Clone 2	9	9	3	3	3	clone_test	m-file	Clone 0	Clone 1

図 6: ConQAT による出力例

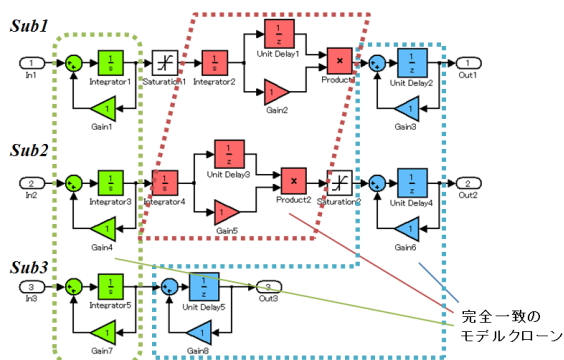


図 7: 完全一致のモデルクローン

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 ii 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、図 4 の Simulink モデルからグラフ構造データを得る過程を、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 8 に示す。

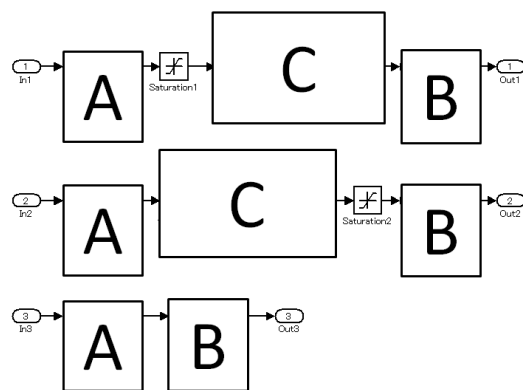


図 8: 単一のノードによる置き換え

3.2.2 ②-ii 完全一致のモデルクローン以外のブロックの除去

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 8 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 9 に示す。

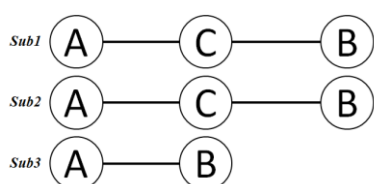


図 9: 完全一致モデルクローン以外のブロックの除去

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 9 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。

$$\begin{aligned}
 \text{Sub1} &= \begin{matrix} & C & B & A \\ C & (0 & 0 & 1) \\ B & (0 & 0 & 1) \\ A & (1 & 1 & 0) \end{matrix} & \text{Sub2} &= \begin{matrix} & C & B & A \\ C & (0 & 0 & 1) \\ B & (0 & 0 & 1) \\ A & (1 & 1 & 0) \end{matrix} \\
 \text{Sub3} &= \begin{matrix} & C & B \\ C & (0 & 1) \\ B & (1 & 0) \end{matrix}
 \end{aligned}$$

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、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。そこで、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で、同一のグラフを表す隣接行列が複数存在することがあるため、正準化する。
- iv. 頻度計算: 正準化された隣接行列を多頻度グラフパターン候補としてそれぞれ支持度を求め、最低支持度および最低サイズを満たす隣接行列を、非完全一致モデルクローンに対応する多頻度グラフパターンとする。
- 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、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。

$$\begin{matrix} & A & & B & & C \\ A & (0) & & B & (0) & C & (0) \\ & & & & & & \\ C & \begin{matrix} C & A \\ (0 & 1) \end{matrix} & & C & \begin{matrix} C & B \\ (0 & 1) \end{matrix} & & C & \begin{matrix} C & B & A \\ (0 & 1 & 1) \\ B & (1 & 0 & 0) \\ A & (1 & 0 & 0) \end{matrix} \\ A & \begin{matrix} (1 & 0) \end{matrix} & & B & \begin{matrix} (1 & 0) \end{matrix} & & A & \begin{matrix} (1 & 0 & 0) \end{matrix} \end{matrix}$$

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。

$$\begin{matrix} & C & B & A \\ C & (0 & 1 & 1) \\ B & (1 & 0 & 0) \\ A & (1 & 0 & 0) \end{matrix}$$

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 9 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 10 に示す.

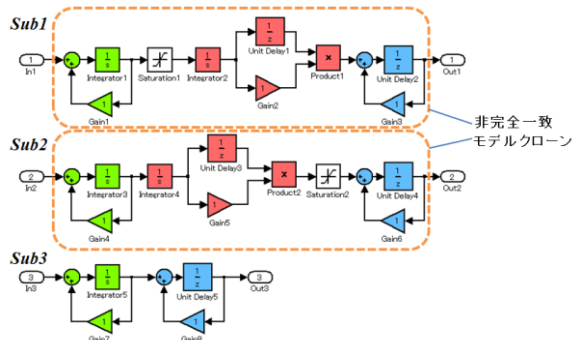


図 10: 非完全一致のモデルクローン

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, 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[12]を用いた.

以降において, 一定規模の同一モデル内でクローンを検出する場合と, 複数のモデル間でクローンを検出する場合のそれぞれについて, 提案手法を適用し, その有効性を検証する.

4.1 単一モデルにおけるクローン検出

単一モデルに対する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 11 に, 各 Subsystem の内容を詳細図として図 1.2 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(図 13)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 12 に示す. また結果をまとめた表を図 14 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・拡張方法の識別が容易になると考えられる.

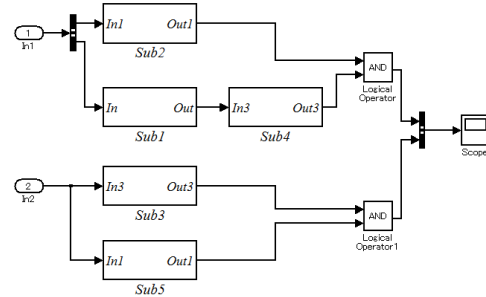


図 11: 入力 Simulink モデルの全体図

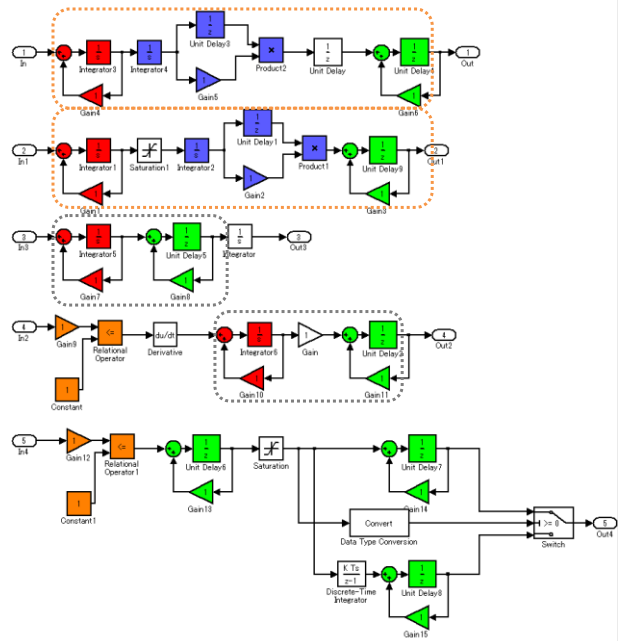


図 12: モデルクローンの構造と位置(色分け: 完全一致クローン, 点線囲い: 非完全一致クローン)

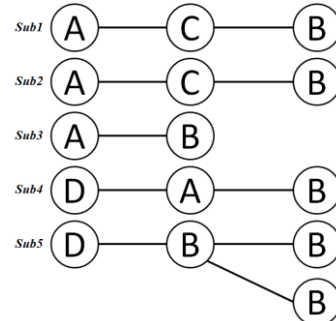


図 13: 完全一致モデルクローンの位置関係グラフ

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 14: モデルクローンの検出結果

4.2 異なるモデル間におけるクローン検出

複数モデルに対する提案手法の有効性を確認するため、PID 制御を行うという共通性を持った Simulink モデルをそれぞれ異なる Web サイトから合計 3 個収集し、提案手法を適用した。その結果、図 15 の点線に示すように、PID 制御を行なっている構造(左側)と、線形伝達関数の実行と表示を行なっているブロックを組み合わせた構造(右側)を、それぞれ完全一致のモデルクローンとして検出し、それらを含む、システムの本質的な部分をより大きな構造(点線内)である非完全一致のモデルクローンとして検出することができた。

このようなクローンをいわば設計上のパターンとして識別することで、これらの 3 つのモデルの保守性を向上させることができると考えられる。例えば、このクローンが担う機能以外の拡張を各モデルに施したい場合は、当該クローンの個所を識別済みであるため修正や拡張の必要な箇所の迅速な絞り込みができる可能性がある。あるいは逆に、クローンが担う機能の修正が必要な場合は、これらの 3 つのモデルを資産として扱う場合に、それぞれについて修正が必要な箇所および修正方法を容易に特定可能となると考えられる。

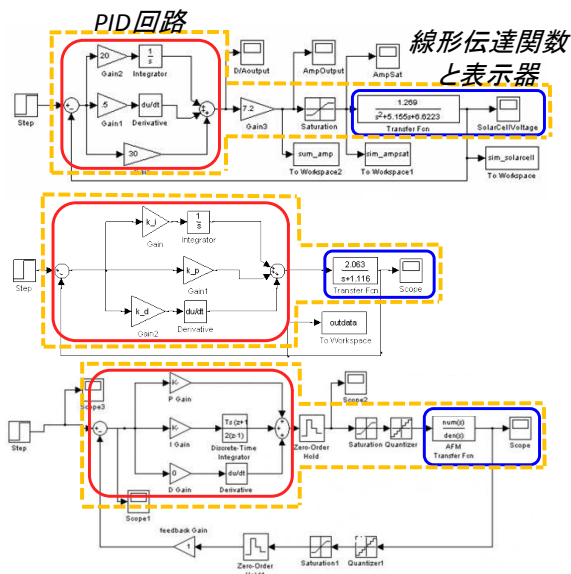


図 15: 3 つのモデルとモデルクローンの構造と位置

5. 関連研究

Bakr Al-Batran らは、既存のモデルクローン検出に検する研究[5][7][8][9]が、構造の一致のみをクローンとして定義している点を問題として指摘し、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比較的小規模な部分を意味的モデルクローンと定義し、これを検出する手法を提案している[2]. 同手法の扱う意味的モデルクローンの規模は基本的に、完全一致のモデルクローンと同等であり、完全一致のモデルクローンよりも大きな粒度のクローンを検出しない。これに対して提案手法は、より大きな概ね構造の一致した部分を非完全一致クローンとして検出する。今後、提案手法の最初のステップ①で検出する完全一致モデルクローンの代わりに、意味的モデルクローンを扱うことで、より広範な非完全一致かつ意味的にも近いクローンを扱うことを検討したい。

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[3]. AGM アルゴリズムが隣接行列のサイズごとにグラフを幅優先で探索するのに対し、Nam H.Pham らの提案する完全一致のモデルクローン検出アルゴリズムではモデル上の特定のノードから開始し、そこから拡大できる構造を他箇所でも検出できるかを調べるという深さ優先探索のアプローチをとっている。Nam H.Pham らの提案する非完全一致のモデルクローン検出アルゴリズムでは、モデルを頂点と辺という要素で捉え、隣接行列の表現を使うというグラフマイニング一般的な手法を用いず、辺をベクトルとして表し、ベクトルの合成で類似度を評価するというアプローチをとっている。しかし、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、比較的小規模なものにとどまる。その点で本論文におけるより大きな粒度の非完全一致のモデルクローンとは異なる。今後は両手法の併用により、大規模かつ非完全一致のモデルクローンを広範に検出することを検討したい。

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なかった比較的大きく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。

提案手法においては最初に完全一致のモデルクロー

ーンを検出し、完全一致のモデルクローンを構成するブロック群を単一のノードに置き換えると同時に、その他のブロックを一時的に除去して、完全一致のモデルクローンの位置関係をグラフ構造で整理した。次に、グラフマイニングのアルゴリズムである AGM アルゴリズムを利用して非完全一致のモデルクローンとなる多頻度グラフを検出し、そのグラフを元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検出した。適用実験として、一定規模の単一モデルおよび複数の実際の制御モデルからそれぞれ、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。提案手法の適用により効率的に、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ことにより、検出対象モデル群の全体の理解や、以降の保守をより容易なものとする事が期待できる。

今後の展望として、より大規模かつ実用の制御システム・ソフトウェアのモデルを対象に適用実験を行い、提案手法の実用性を検証することを予定している。

また、完全一致であるがゆえに類似度が 100% であることが自明な完全一致のモデルクローンとは異なり、提案手法が扱う非完全一致のモデルクローンにおいては、その類似度を定量的に評価することが、将来の保守の支援にあたり重要と考えられる。モデルクローンを構成しているブロック群の一致度合いや、構成しているブロックの種類などの情報を元に、類似度を定量的に評価するメトリクスを定義することで、検出した非完全一致のモデルクローンをどのように扱うべきか、指標とすることができると考えられる。

さらに我々は、非完全一致のモデルクローンを検出する範囲として、Simulink モデルの Subsystem を利用した。中規模以上の Simulink モデルにおいては、Subsystem によるモジュール化は頻繁に行われており、機能としてのまとまりが強いブロック群としてみなすことができる。しかし実際には、機能や、まとまることで 1 つの式による演算を実現しているようなすべてのブロック群が、すべて Subsystem としてまとめられているわけではない。Simulink モデルのブロックを適切にクラスタリングする技術と組み合わせることで、設計上の意味合いがより強く反映された単位で、非完全一致のモデルクローンを切り出すことができるようになると考えられる。

参考文献

- [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] Simulink Library - ConQAT http://conqat.in.tum.de/index.php/Simulink_Library
- [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] 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.
- [9] Al-Batran, B. "Model-Based Clone Detection Using Normal Forms." Master's thesis, Technische Universitat Munchen, 2011
- [10] 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.
- [11] Agrawal, R. and Srikant, R. "Fast Algorithms for Mining Association Rules, in Bocca, J. B., Jarke, M., and Zaniolo, C. eds.", In Proc. of the 20th Very Large Data Bases Conference, pp.487-499, 1994.
- [12] MUSASHI, <http://musashi.sourceforge.jp/>