

MBD中部カンファレンス

# MATLAB環境におけるリファクタ リングツールの開発について

サイバネットシステム株式会社  
システムCAE事業部  
平坂 昭人

サイバネットシステム株式会社



# はじめに

類似するサブシステム  
をコピー&修正



モデルを作成、改良を繰り返し進めていると  
気づいたときには

- ・ 同じサブシステムを複製
- ・ 内部のパラメータのみ変更して複製

といったサブシステムを作ってしまった  
ことがありませんか？

## リファクタリング手法

折角、作ったモデルの動作を変えずにライブラリ化し、内部構造を整理したい！

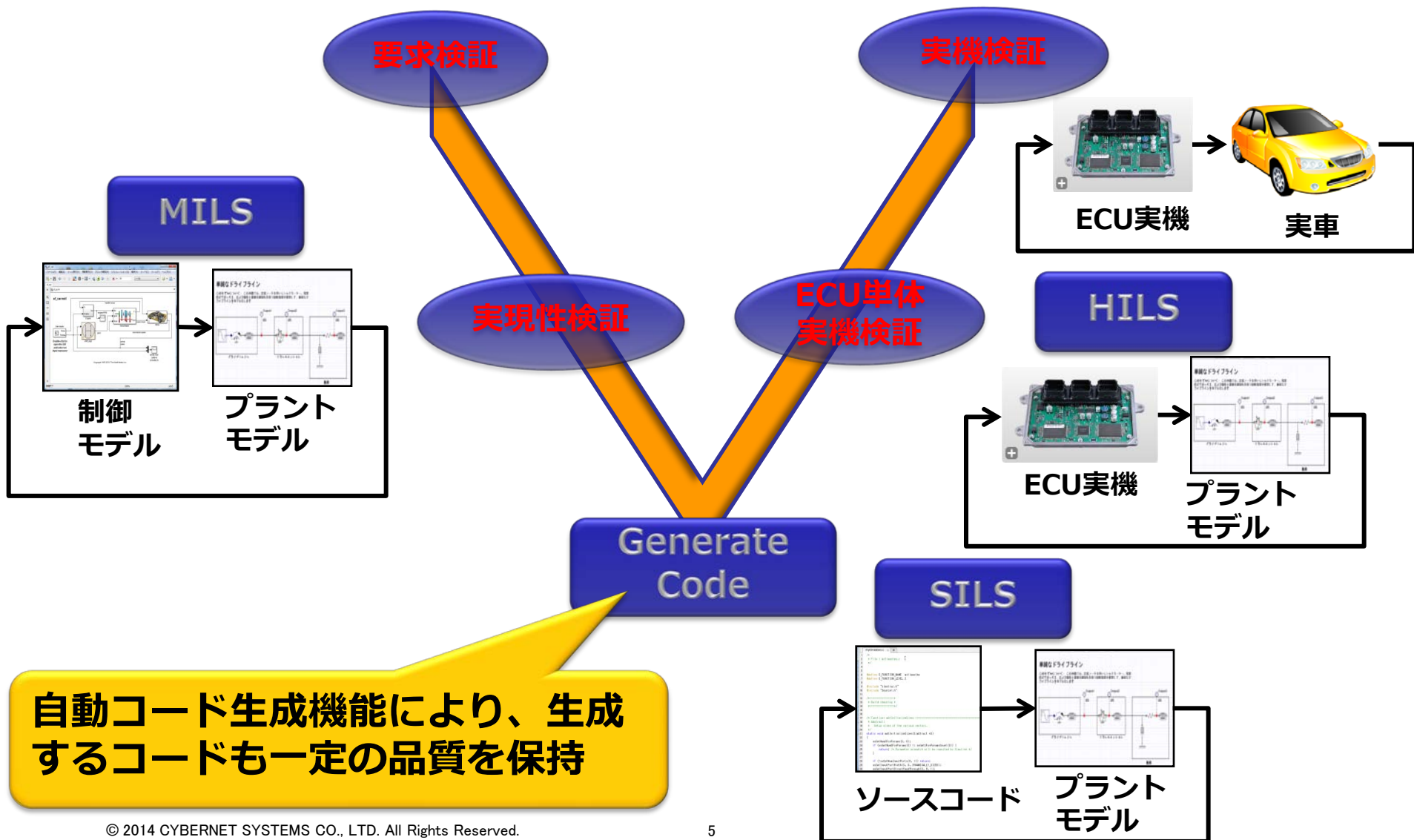


- **リファクタリング (refactoring) とはコンピュータプログラミングにおいて、プログラムの外部から見た動作を変えずにソースコードの内部構造を整理すること。**いくつかのリファクタリング手法の総称としても使われる。十分に確立された技術とはいえ、**「リファクタリング」**の語にも厳密な定義があるわけではない。  
※Wikipedia引用

## 本公演の流れ

- 背景/問題点の共有
- リファクタリングツールの開発
  - リファクタリング手法
  - ツール開発
  - 効果
- まとめ

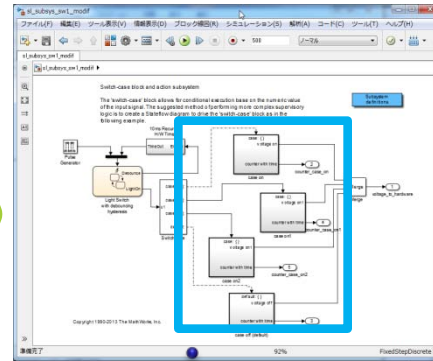
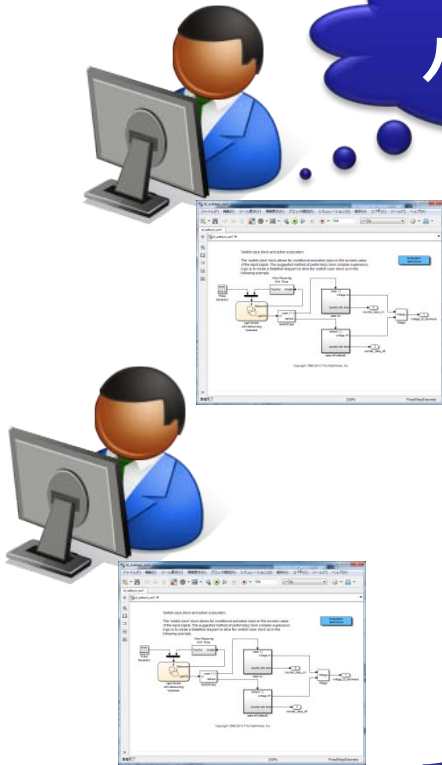
# 背景



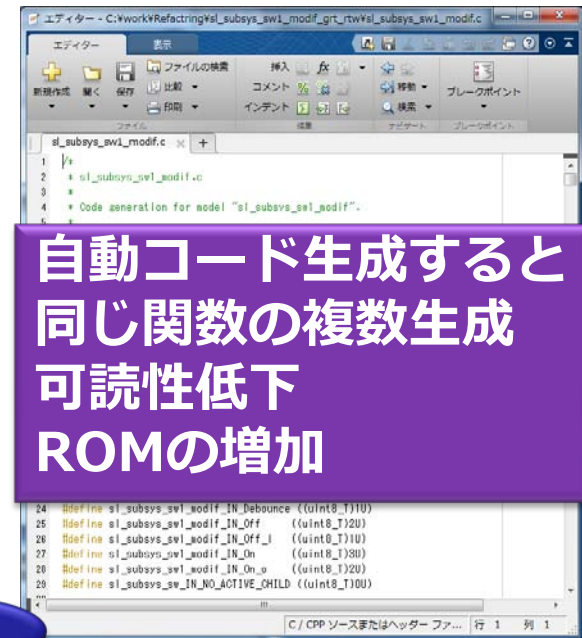
自動コード生成機能により、生成するコードも一定の品質を保持

# 背景

設計変更  
バグフィックス



自動コード生成すると  
同じ関数の複数生成  
可読性低下  
ROMの増加



同一(類似)の  
サブシステムが存在

## 問題点

	シミュレーション (モデル)	コード生成
可読性	ブロック線図の可読性が悪い。	類似する関数が複数でき、可読性が悪い。
生産性	ライブラリ化されていない場合は、複製や修正が必要になり、工数がかかる。	ソースコードが増え、マイコンへの実装に障害となる。

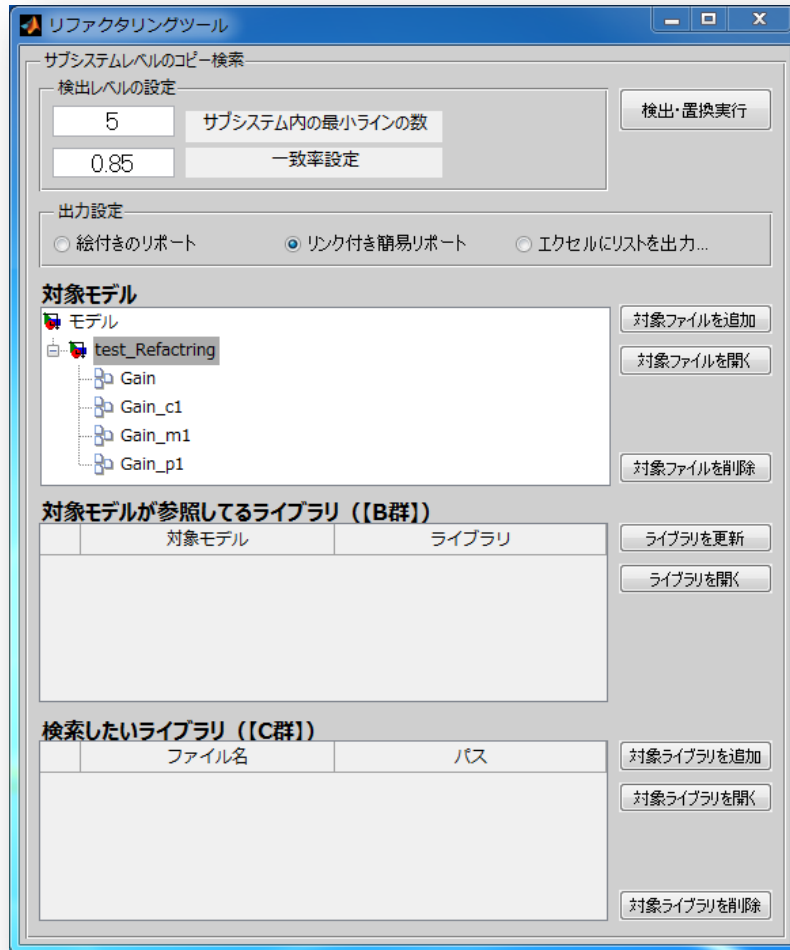
大規模モデルでは手動での修正作業は煩雑でミスが起こりやすい問題点



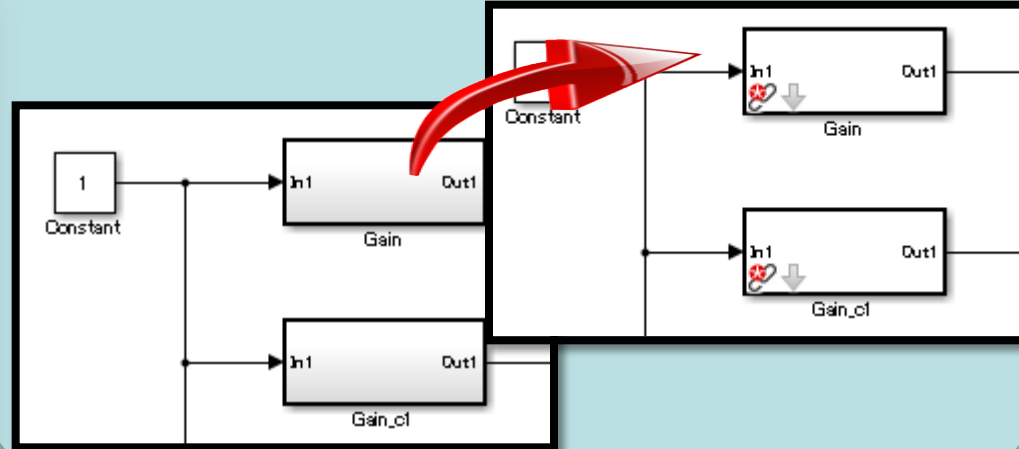
このような問題を効率的に解決するために、リファクタリングツールの開発しました。

# ツール開発

## Refactoring Tool



## 共通サブシステムの探索と自動置換機能



## レポートの出力機能

### クローン自動検索・置換

liang  
22-Aug-2013 202721

目次

[クローン自動置換情報](#)  
[モデルと\[\[B群\]\]\[C群\]の部分一致のサブシステムリスト](#)  
[モデルと\[\[B群\]\]\[C群\]の完全一致のサブシステムリスト](#)  
[モデルと\[\[B群\]\]\[C群\]の部分一致のサブシステムの置換情報](#)

表目次

- 1 部分一致のリスト
- 2 完全一致のリスト1
- 3 完全一致のリスト2
- 4 完全一致のリスト3
- 5 完全一致のリスト4

HTML形式  
Excel形式



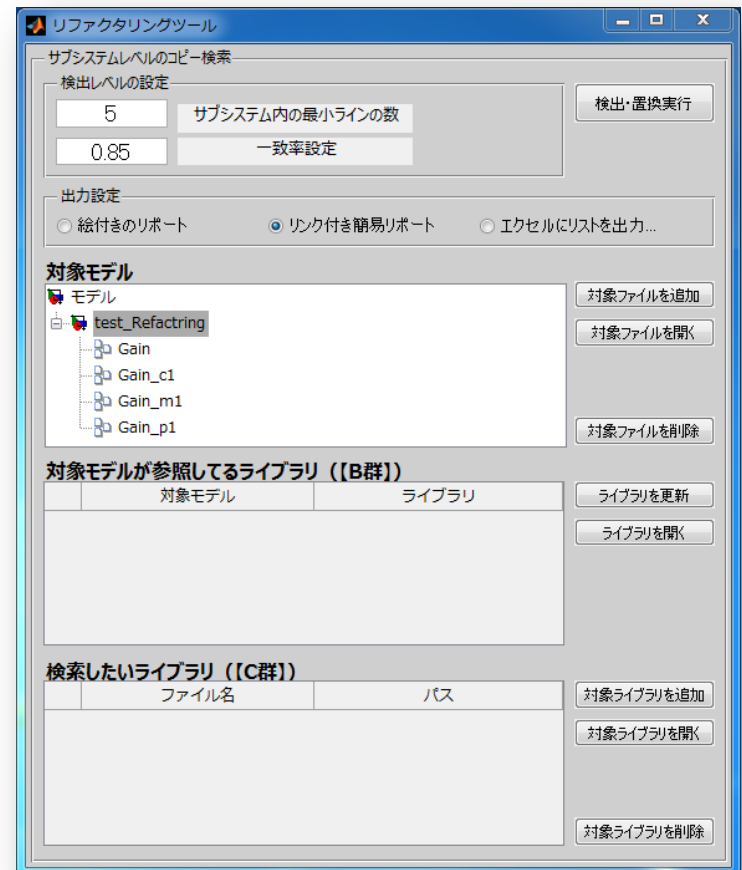
## ツール開発：リファクタリング手法

	① サブシステムの境界を無視して、共通的に使われるブロックの組み合わせを調べる	② サブシステム単位で共通サブシステムを調べる
<b>GOOD</b>	総当りで細かい組み合わせを探索可能	<ul style="list-style-type: none"> <li>•サブシステム単位のため、自動的共通関数化しやすい。</li> <li>•比較的短時間での探索が可能。</li> </ul>
<b>BAD</b>	<ul style="list-style-type: none"> <li>•検出時間が爆発的に増加</li> <li>•共通ライブラリとするブロックの連結サイズを人が決める</li> </ul>	①に比べて、共通化できる対象の漏れが多い

**Simulinkの抽象度を利用し、高速に検索できるような工夫を考慮しつつ②の手法でツール開発をしました。**

## ツール開発

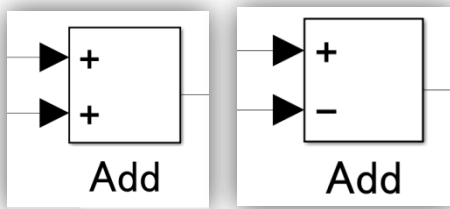
- 開発の工夫について
  - モデル情報の取得およびその管理
  - 一致率の算出
  - モデルの置き換え方



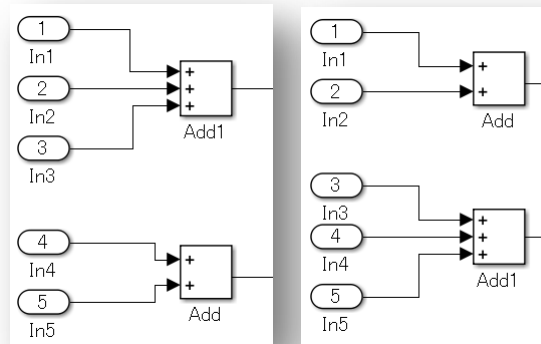
## ツール開発 : モデル情報と一致性の判断

### 本ツール開発の最大の難しさ

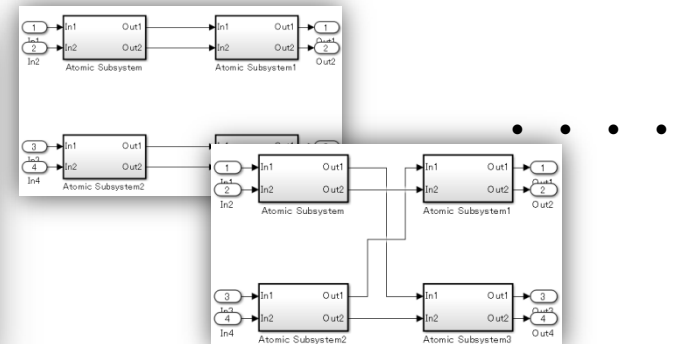
各サブシステムが同一かどうかを判断するために  
どのような情報を取得すればよいか、またどう判断するか



ブロックは同じでも  
演算処理が異なる



使用ブロックは同じでも  
ブロックの接続が異なる



ポート番号の接  
続が異なる

1. 不特定のモデルを判断するため構成ブロックのデータ化
2. そのデータを利用して、サブシステム間の同一性を数値化

# ツール開発 : モデル情報の取得およびその管理

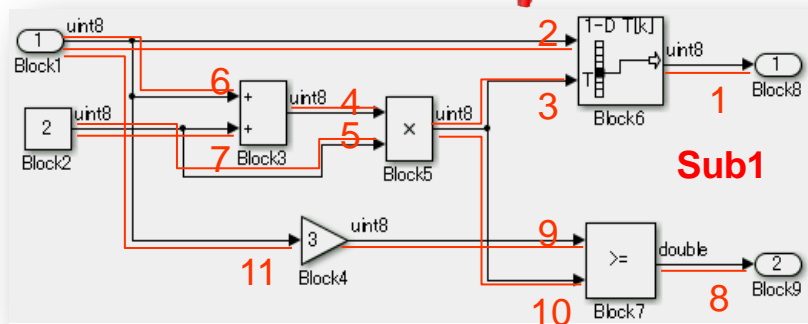
検出対象のサブシステム内部の情報を以下の標準関数を利用し取得します。

find\_system : システム内のブロック、ライン、端子等の検索

get : Handle Graphics オブジェクト プロパティのクエリの取得

get\_param : システムとブロックのパラメーター値を取得

主に、以下の情報を構築しモデル状態をデータ化します。



ブロックタイプID			ポート接続ID	
ID	BlockType (再構築的な)	Block Name	ID	ConnectType
1	Constant	Block2	1	1→1
2	Gain	Block4	2	1→2
...				
11	RelationalOperator(==)			
12	RelationalOperator(~=)			

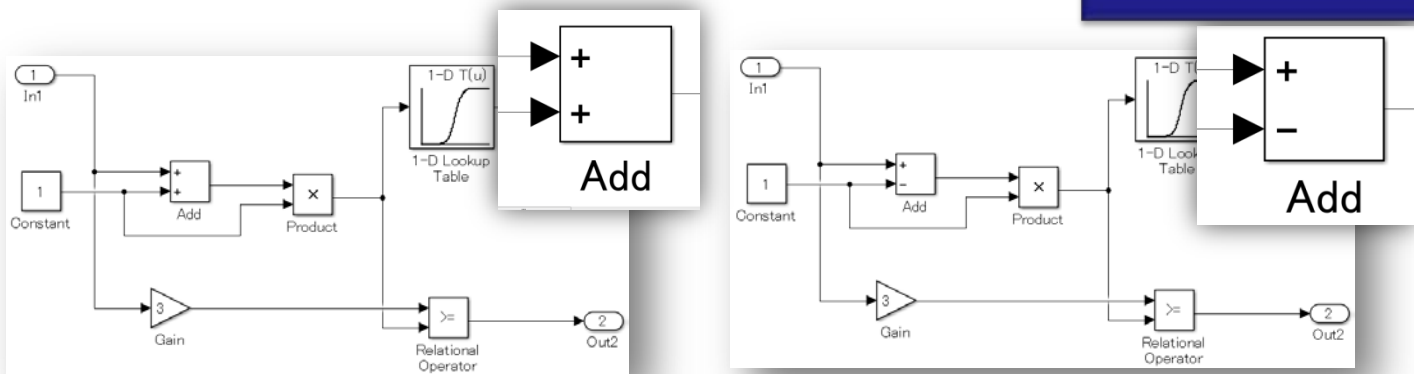
ブロック名ID		ブロックの接続情報	
ID	BlockName	BlockName	接続情報
1	Block1	Block1	
2	Block2	Block2	
...		...	
8	Block8	Block8	6 8 4 5 1
9	Block9	Block9	7 9 8 6 1

# ツール開発：一致率の算出

サブシステム間の同一性を一致率として数値化  
 ⇒完全一致するサブシステムだけでなく、類似するサブシステムも評価可能

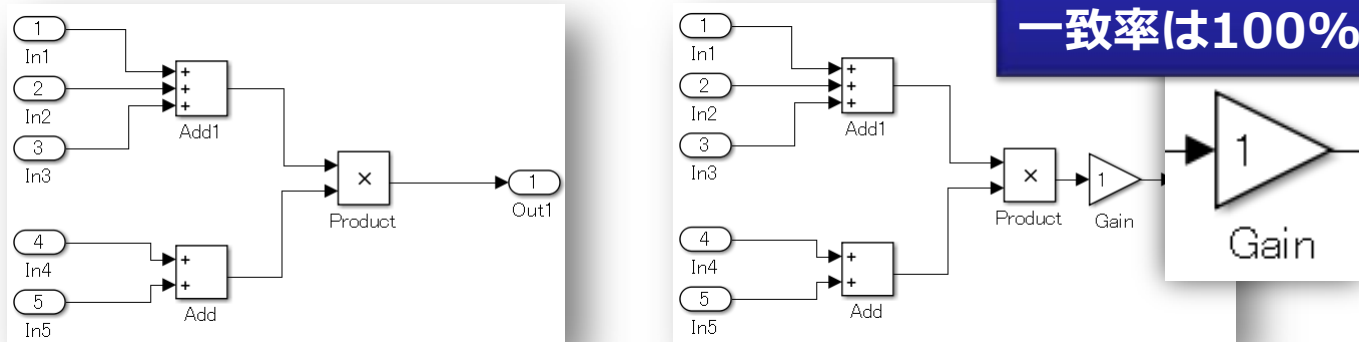
例：Addブロックの演算符号も考慮

部分一致  
一致率は70%



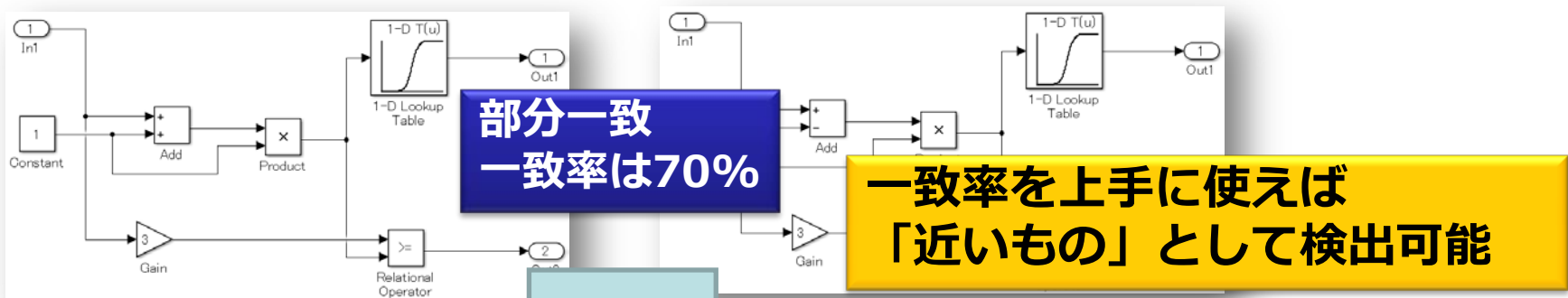
例：ゲイン[1]のブロックを除外

完全一致  
一致率は100%



# BADに対する改善

	②サブシステム単位で共通サブシステムを調べる
<b>BAD</b>	①に比べて、共通化できる対象の漏れが多い



## レポートの出力機能

クローン自動検索・置換

liang

22-Aug-2013 2027:21

---

目次

[クローン自動置換情報](#)

[モデルと【e群】\(c群\)の部分一致のサブシステムリスト](#)

[モデルと【e群】\(c群\)の完全一致のサブシステムリスト](#)

[モデルと【e群】\(c群\)の完全一致のサブシステムの置換情報](#)

表目次

1. 部分一致のリスト
2. 完全一致のリスト-1
3. 完全一致のリスト-2
4. 完全一致のリスト-3

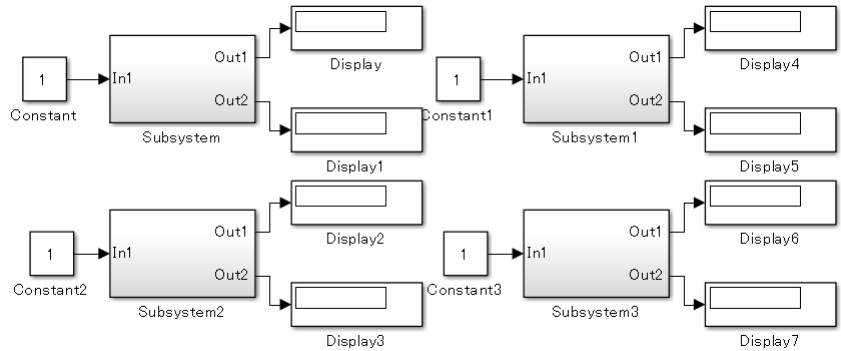
### 絵付のHTMLを出力



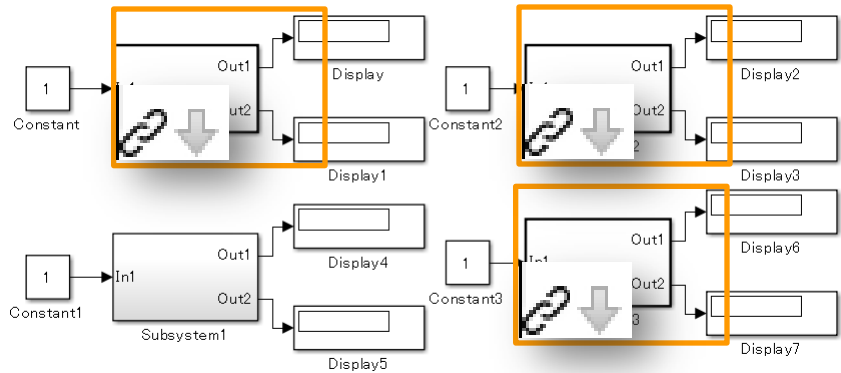
- 一定の一致率以上のものはレポート出力
- 人の確認時間は激減。

# ツール開発：モデルの置き換え方

1. 既存のライブラリと一致するサブシステムは自動的に置換
2. 1.以外の共通サブシステムは置換するためのMファイルを自動生成  
→ユーザが適宜カスタマイズ可能



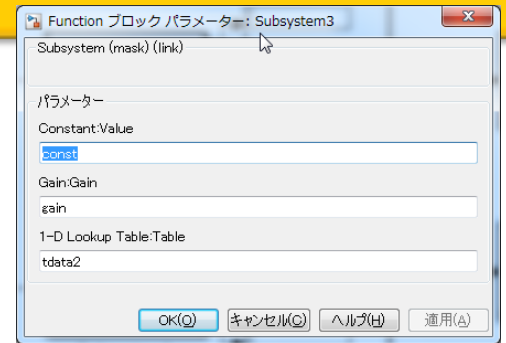
## リファクタリング後 共通サブシステムはライブラリにリンク



```

test_Refactring_Clona_SemiAuto.m
このファイルは書式化されたドキュメントへのパブリッシュが可能です。詳細については、ビデオまたはヘルプのパブリッシュに
1 function test_Refactring_Clona_SemiAuto()
2 % In this file, subsystems that will be added to library and replaced are logged.
3 %% ここからは処理されるサブシステム情報です。
4 %-----
5 %% * 最初のモデル
6 %% test_Refactring
7 %%
8 %% * 対象サブシステム
9 %% すべてのサブシステム
10 %% * 対象サブシステム
11 %%
12 %% * ブロックの自動置換情報
13 %% C:\Work\Refactring\Test Refactring ClonaReport Auto.xls
    
```

調整可能なパラメータがある場合はマスク化も行います。



## 開発ツールの効果

実際のモデル規模による検証結果を下表に示します。

<b>モデル概要</b>	総ブロック数	17597
	サブシステム数	1551
	共通サブシステム数 パターン	102(内99個置換) 37(内24個置換)
<b>処理時間 [sec]</b>		既存ライブラリの自動置換 : 97.7 既存ライブラリ以外の置換 : 102.3

**手作業でのモデル修正に比べると  
非常に効率的な結果**

WinOS/32bit  
メモリ : 2GB  
CPU : 3.16GHz



## まとめ

### 問題点

冗長なモデルは可読性や生産性を低下させるため、モデルの整理が必要

### 工夫

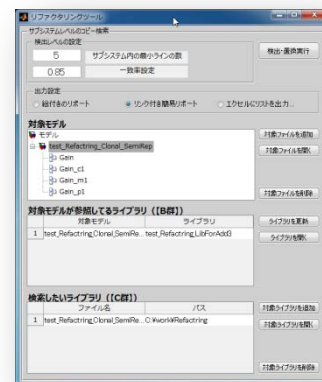
リファクタリング手法を利用し、自動的に共通サブシステムを探索し、モデルを整理するツールを開発

### 結果

元の機能を損なわず、効率的にモデルの整理が可能となり、開発環境が向上

### 謝辞

本ツール開発はアイシンAW 久保様のアイデアをもとに開発を行っています。このような機会をいただきありがとうございます。



ご清聴ありがとうございました。