

JMAAB制御モデリングガイドラインの 現状と将来

2016年9月16日

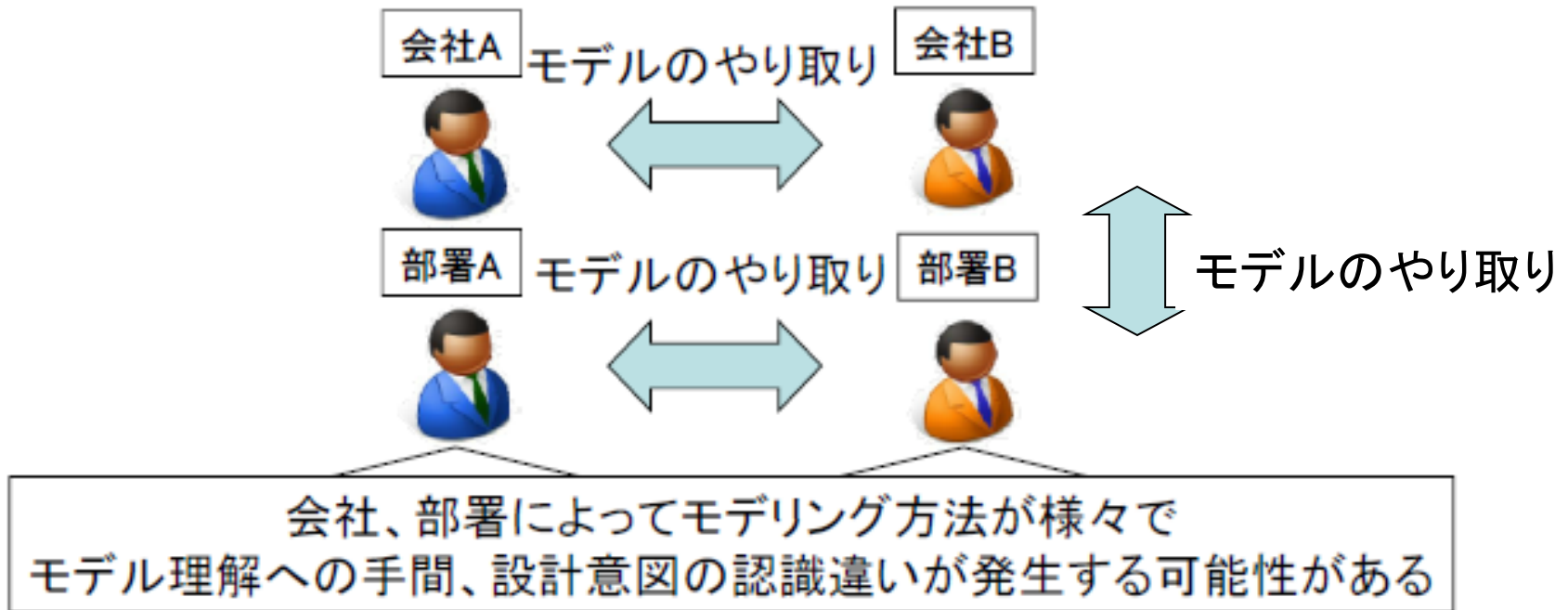
アイシン・コムクルーズ株式会社
久保孝行

下記の3つのパートを説明します。

- ガイドラインの歴史とVer4の特徴
- Ver5.0に向けた制御モデリングガイドラインWG活動概要
- 新ルール「ここが変わる」Ver5.0での変更内容の説明
- ガイドラインの将来

背景

ソフトウェア開発において、Simulink を使ったモデルベース開発が行われるようになってきた。



目的

業界スタンダードを
作り上げる

モデルの書き方を統一し、異なる設計者同士で
共通の理解を容易に得られる

CONTROL ALGORITHM MODELING GUIDELINES USING MATLAB[®], Simulink[®], and Stateflow[®]

CONTROL ALGORITHM MODELING GUIDELINES USING MATLAB[®], Simulink[®], and Stateflow[®] Version 4.0 (日本語版)

Japan MBD Automotive Advisory Board (JMAAB)
2015年03月31日

4.3.7. jc_0121 : Sum ブロックの使用法

ID : タイトル	jc_0121 : Sum ブロックの使用法
製作者	池田
範囲	JMAAB
MATLABバージョン	ALL
前提条件	Sum ブロックの使用条件 <ul style="list-style-type: none"> 四角形を使用します。 入力信号が異なるようにリサイズ調整します。 第一入力の符号は+とします。

記述内容

【正】

【誤1】

【誤2】

1) 入力角が 90°, 180°, 270°以外から入力されています。
2) 符号が異なり判別できません。

【誤3】

1) 最初の入力角が-の符号を使っています。これは、可読性を損ないます。

フィードバックループでは内部を使用できません。使用する場合は、下記3つの条件を守る必要があります。

- 入力数は 2~3 つまでにしてください。

- 入力は 90°, 180°, 270°の位置に合わせます。
- 出力は 0°の位置に合わせます。

フィードバックループの場合、先述の四角形でも第一入力の符号に+を使用可能です。

【正】

その他の記述例

誤りの理由
丸型を使用する場合、入力は 90°, 180°, 270°以外の角度は使用できません。

標準
MSTRAC SLSF 010A

最終更新
V4.0

自動車用の制御装置のコントローラモデルを運用するうえで、Simulink/Stateflowモデルの記述についてルールを規定したもの

JMAABのワーキング活動で作成したガイドラインを「JMAABスタイルガイドライン」と記載しています。

Ver1.0

1. アイシン精機
 2. ジヤトコ
 3. デンソー
 4. トヨタ自動車
 5. 日産自動車
 6. 日立製作所
 7. 本田技術研究所
 8. マツダ
- サイバネットシステム

Ver2.0

1. アイシンAW
 2. アイシン精機
 3. いすゞ自動車
 4. ジヤトコ
 5. トヨタ自動車
 6. 日立製作所
 7. マツダ
 8. 三菱電機
- サイバネットシステム

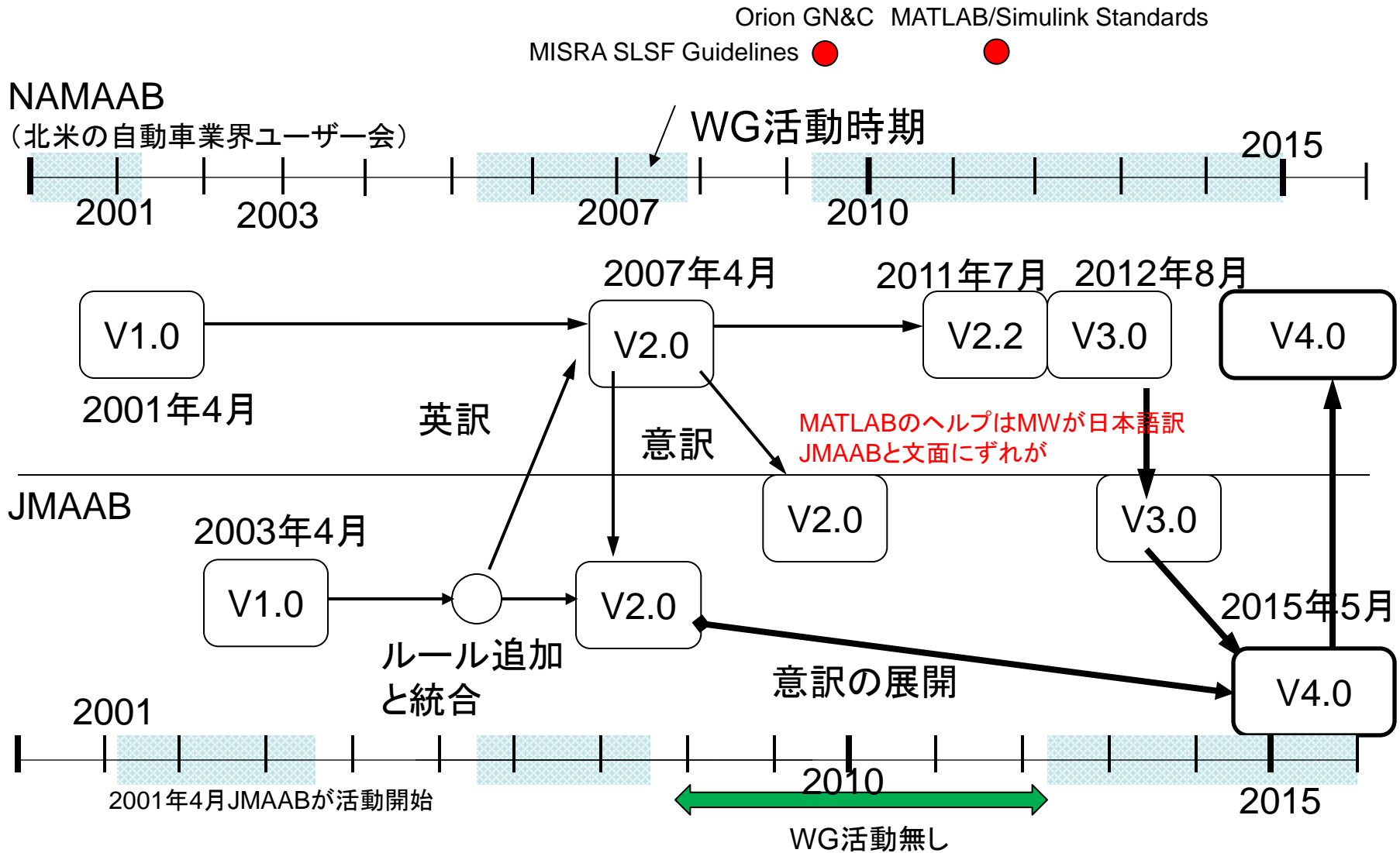
Ver4.0

1. アイシンAW
 2. アイシン精機
 3. いすゞ自動車
 4. オムロンオートモーティブ
エレクトロニクス
 5. カルソニックカンセイ
 6. スズキ
 7. ダイハツ工業
 8. デンソー
 9. トヨタ自動車
 10. 日産自動車
 11. ミツバ
 12. 三菱電機
- マスワークスジャパン

青字: 前回Verからの追加メンバー

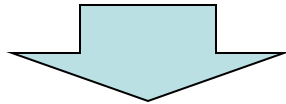
- 五十音順
- 株式会社は省略
- 日立製作所は、現在日立オートモーティブシステムズ

JMAABスタイルガイドライン改定の流れ



NAMAAB側のルール追加の特徴

- ルールではなく、概念が多い。
- ルールの意図が不明。
- 禁則事項が不明確でチェックができない



JMAAB側の方向性

- ・概念はルールではない。
- ・ルールは、一意に解釈でき、自動的に決定可能であるべき
- ・流派は、流派として区別すべき

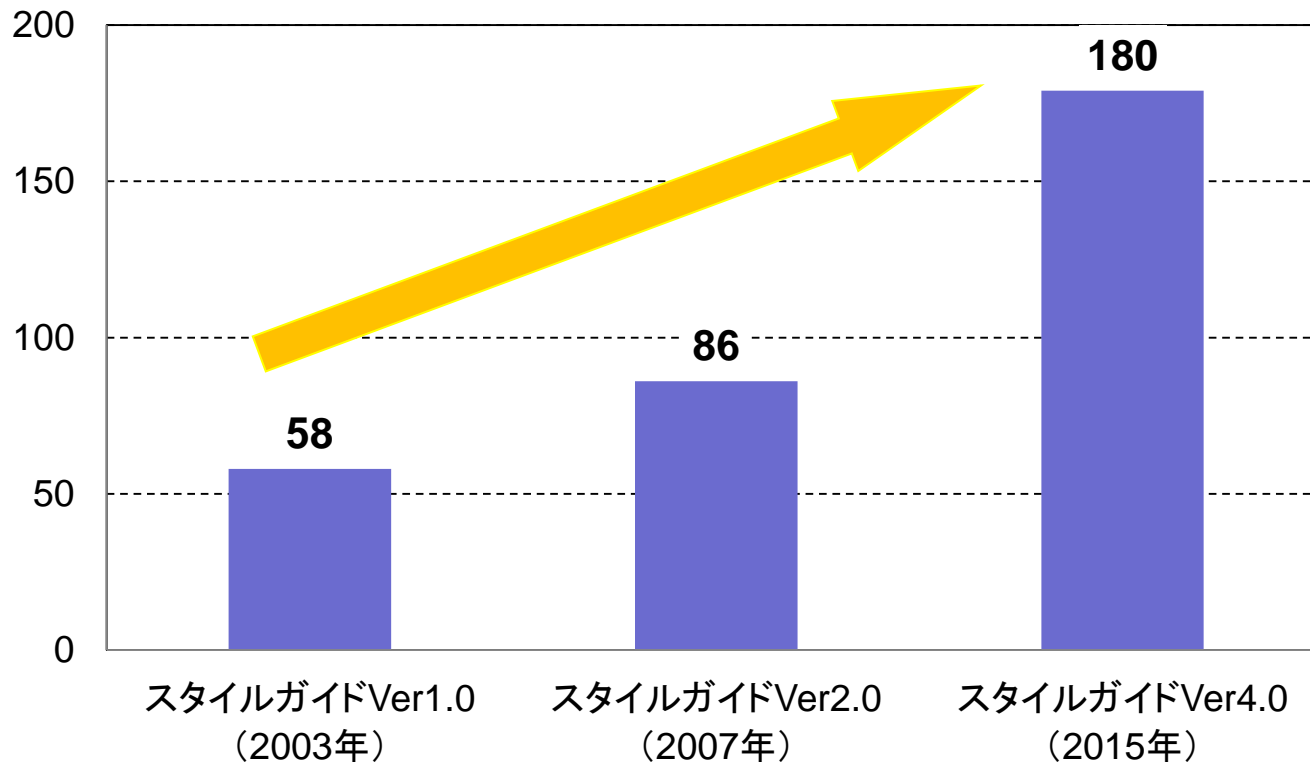
NAMAAB側が、独自に改定するのを任せられない。

JMAAB側に、ガイドラインを改定するWGが必要！

NAMAABからのVer3リリース前後から、JMAAB側で再活動開始。

- 2003年のJMAABスタイルガイドラインV1.0
 - モデル構造に対する考え方を中心にガイドラインが構成される。
 - Data Dictionary: Simulink仕様書で使用されるデータに関する規定がまとめられた。
- 2007年のJMAABスタイルガイドラインV2.0
 - JMAABスタイルガイドラインV1.0をMAABガイドラインに統合した両V1の内容に加えて、新ルールを追加した。
- 2015年のJMAABスタイルガイドラインV4.0
 - ルールの詳細決定をユーザーに委ねるようパラメータを設定した。
 - 他ガイドライン(MISRAやORION)との関連を追加した。
 - ノウハウやヘルプ的な記述はIDを削除し、説明文書として掲載した。
 - 同一の内容が複数個所に掲載されたルールを統合した。

- Ver4.0では、各社からの新規ルール追加や、他団体のルールとの関連付けを行った結果、ルール数が大幅に増加しました。

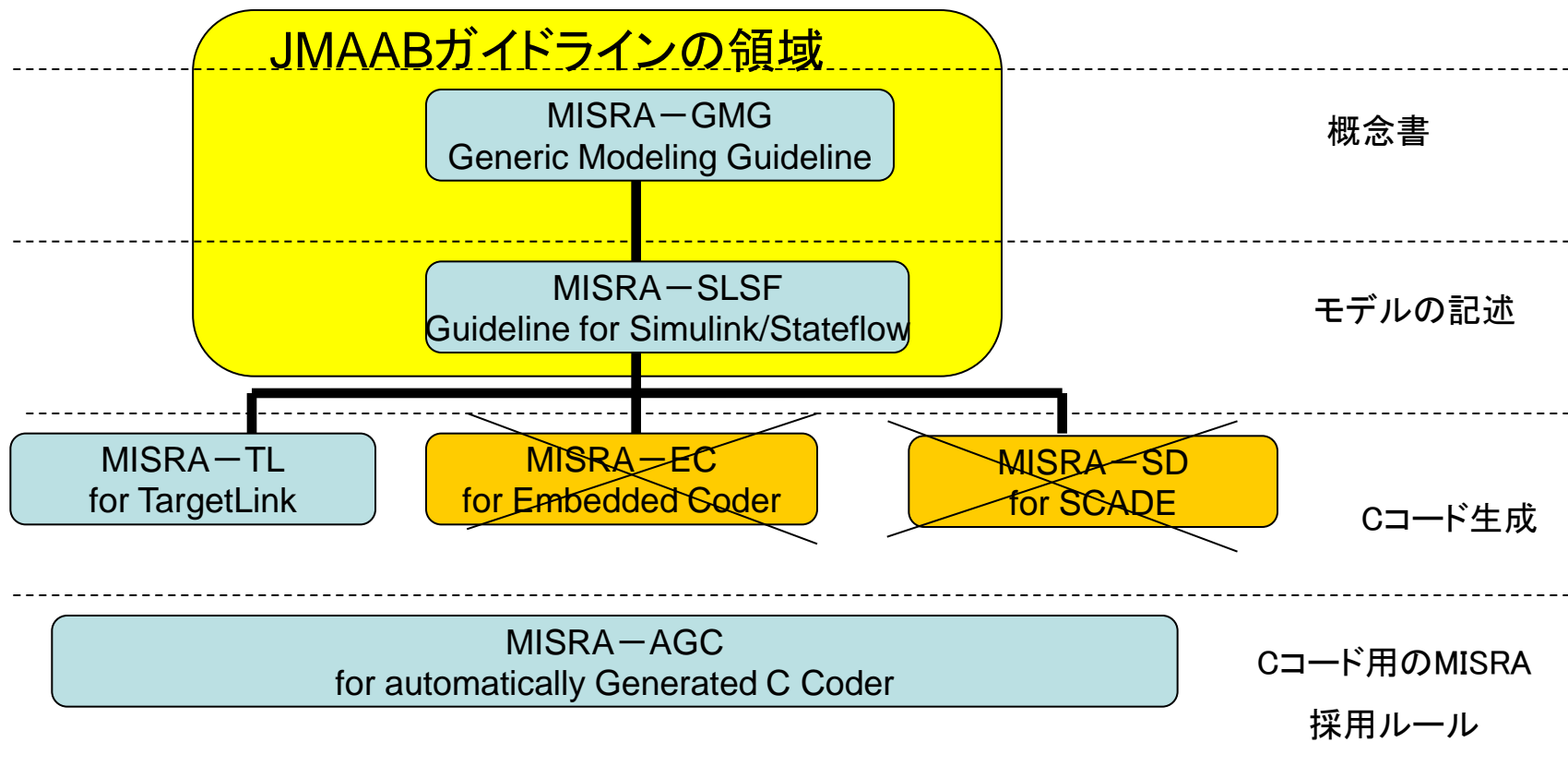


- Simulinkに関連するスタイルガイドが沢山あります。

発行団体名	ガイドライン名	リリース時期
MAAB	MAAB ガイドライン(略名)	2007年7月 Ver2.0 2011年7月 Ver2.2 2012年8月 Ver3.0 2015年3月 Ver4.0
MathWorks	Modeling Guidelines for High-Integrity Systems	2009年9月 Ver1.0 2016年3月 Ver1.13
MathWorks	Modeling Guidelines for Code Generation	2010年9月 Ver1.0 2016年3月 Ver1.11
MISRA	MISRA SLSF Guidelines	2009年5月 Ver1.0
ORION	Orion GN&C MATLAB/Simulink Standards	2011年10月 Ver1.0

Ver4.0作成時の課題

不足しているルールをかき集めるには、多くのガイドラインを見る必要がある。
逆の事が書いてある場合、何を信じれば良いか判断できない。



MISRA SLSF Guidelines Simulink Stateflowのガイドライン

キャリブレーションパラメータあるいは名前付きの定数を含むブロックは、名前を見えるようにするためにサイズ

MISRA AC TL TLの設定

Rationale

MISRA C Rule 5.1 requires that identifiers are distinct in the first 31 characters. TargetLink can be configured to ensure that this rule is adhered to in the generated code.

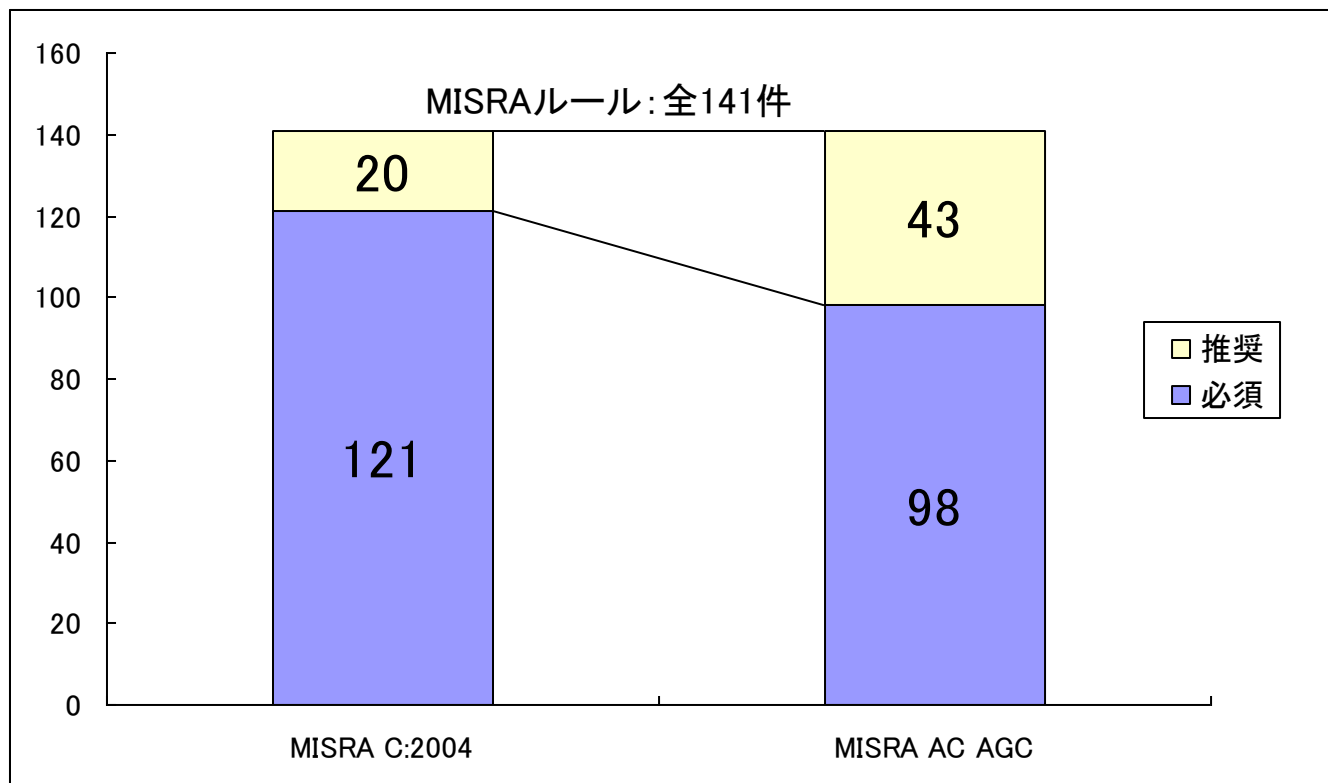
Notes on verification

This rule can be statically checked.

References

None.

Example



MISRA-C:2012はMISRA AC AGCを統合

MISRA-C:2012は、ツールでの検出を想定した文面に変更された。

- Rule(ルール)とDirective(指針)に分離
 - 静的解析ツールの適用を想定しているガイドラインがRule(ルール)に分類、該当しない記述をDirective(指針)に変更
 - JMAAB Ver4以降でルールとそれ以外を分離
- ルール区分をDecidable(決定可能)とUndecidable(決定不能)に分離
 - ルールに準拠してるか否かを論理的に区別できるツールを作り出せるか否かで、Decidable(決定可能)Undecidable(決定不能)に分類
 - JMAAB Ver4以降、あいまいさを排除する取り組みを開始。

Ver4からの変更活動は、MISRA2012の改定時の取り組みと一緒

制御モデリングガイドラインWG 活動概要

JMAABスタイルガイドラインVer5.0

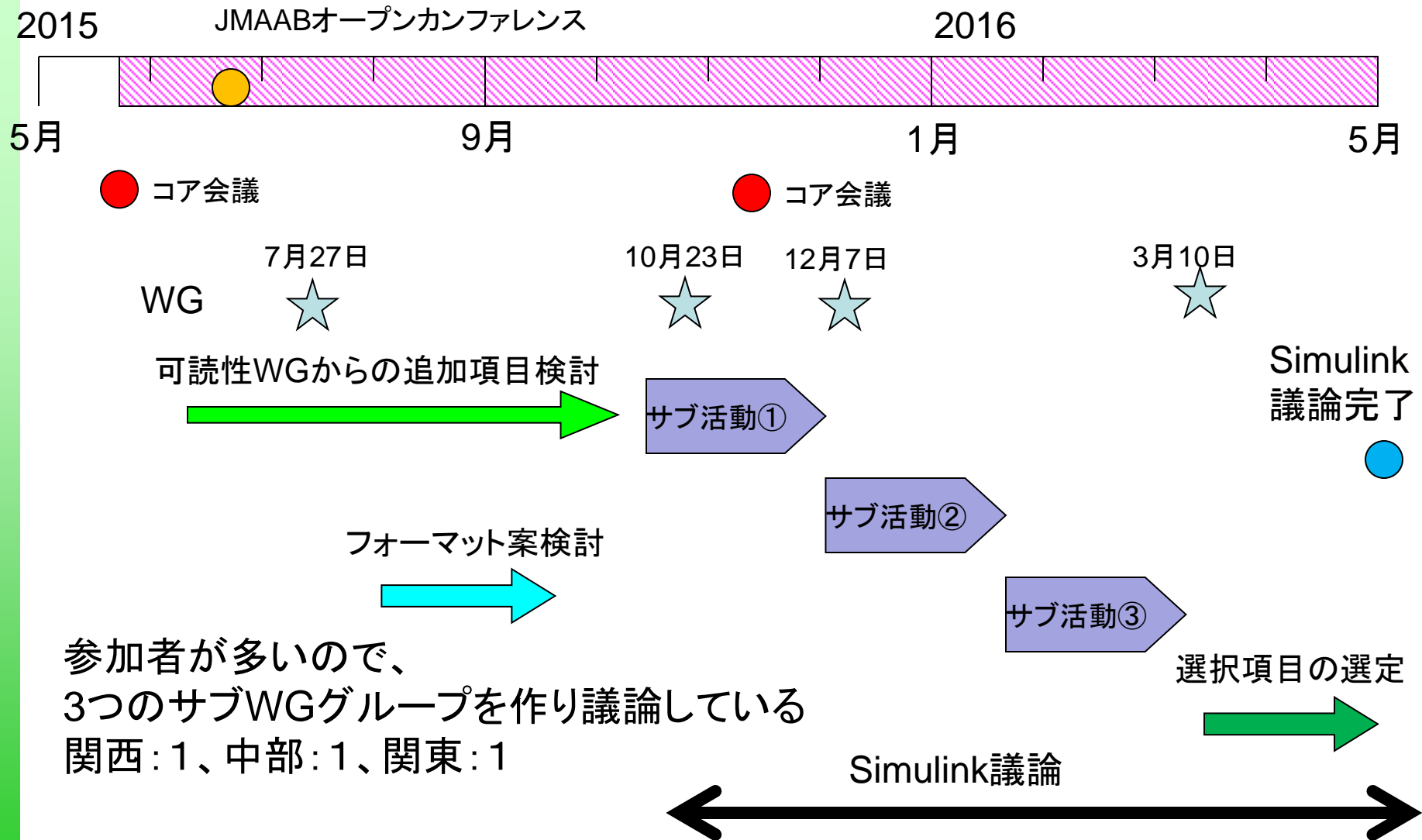
Ver5.0

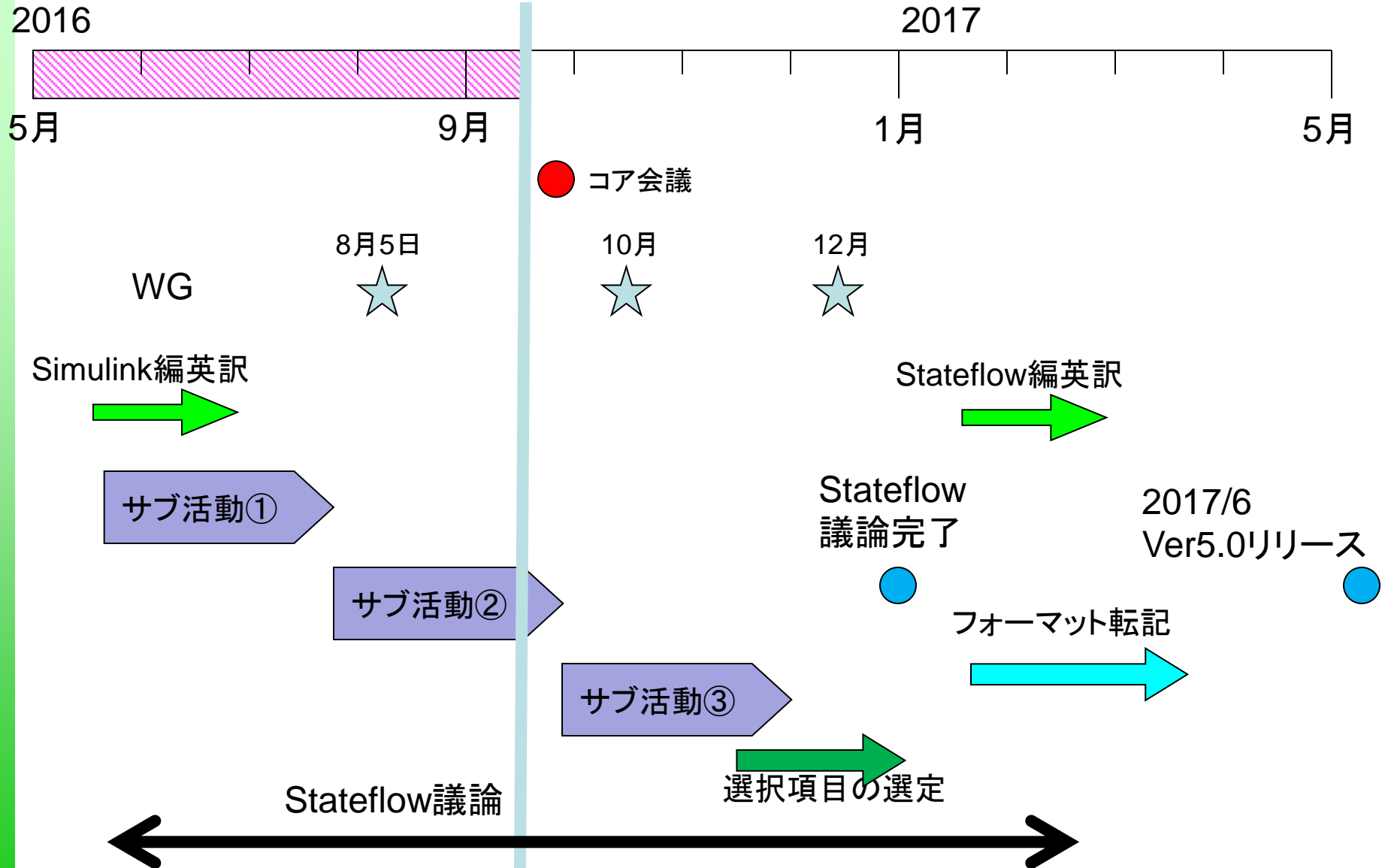
1. アイシンAW
2. アイシン・コムクルーズ
3. アイシン精機
4. いすゞ自動車
5. オムロンオートモーティブ
エレクトロニクス
6. ケーヒン
7. 小松製作所
8. スズキ
9. ダイハツ工業

10. デンソー
11. トヨタ自動車
12. 日立オートモーティブシステムズ
13. 富士通テン
14. マツダ
15. 三菱自動車
16. 三菱電機
17. 両毛システムズ
- マスワークスジャパン

青字: 前回Verからの追加メンバー

- 五十音順
- 株式会社は省略





- ① 複数のルールについてはサブIDを設けることで、ルールごとに採用/不採用が定義でき、重要度も個別に設定できるようにする。
- ② ルールの真意や考え方を明確にし、なぜそのルールを守らなくてはいけないのか、また、守らなかった場合にどんな影響があるのかを掲載する。
- ③ 複数の手順(流派、パターン)がある場合は選択式にする。

これらのコンセプトに従って、作業を進めている。

新ルール「ここが変わる」 Ver5.0での変更内容の説明

- 下記の表記記号を用いることで、曖昧な表現を排除し、記述内容を形式的な表現にしました。

種別	表記	表記例	備考
ブロック名	[]	[Output]	Simulinkライブラリに登録されている名前
パラメーター名	{}	{初期出力}	設定ができるもの(値を持つもの) 2015aの表示名
パラメーター値	""	"0"	チェックボックスの場合は"on"もしくは"off"

【記載例】ID:na_0011

Ver4.0 Gotoブロックではローカル範囲を使用しなければなりません。

Ver5.0 [Goto]の{タグの可視性}は"ローカル"にします。

- jc_0121:加減算ブロック(※)の使用方法(Ver5.0)
 - 加減算ブロックの{アイコン形状}は、“四角形”を使用します。ただし、フィードバックループの場合は、{アイコン形状}に“丸型”を使用できません。
 - 加減算ブロックの第一入力の符号は“+”とします。ただし、フィードバックループの場合は、第一入力の符号に“-”を使用できます。
 - 加減算ブロックの入力数は2つまでとします。
- これらのルールは、一つの意図から発生しているか？
- 全てのルールの重要度は同じなのか？

jc_0121:加減算ブロック(※)の使用方法(Ver5.0)	
ルール	加減算ブロックの{アイコン形状}は、“四角形”を使用します。ただし、フィードバックループの場合は、{アイコン形状}に“丸型”を使用できます。
	根拠 データフローを左から右にする事ができ、可読性が向上します。フィードバックループの場合に円形を使用すると、ループ処理が明確になります。
ルール	加減算ブロックの第一入力の符号は“+”とします。ただし、フィードバックループの場合は、第一入力の符号に“-”を使用できます。
	根拠 第一入力の符号が統一される事で、制御仕様の可読性が向上します。
ルール	加減算ブロックの入力数は2つまでとします。
	根拠 演算順序を明確に規定する事ができます。 (コード生成の設計検討で、オーバーフローの検証をやり易くするなど の意図がある。)

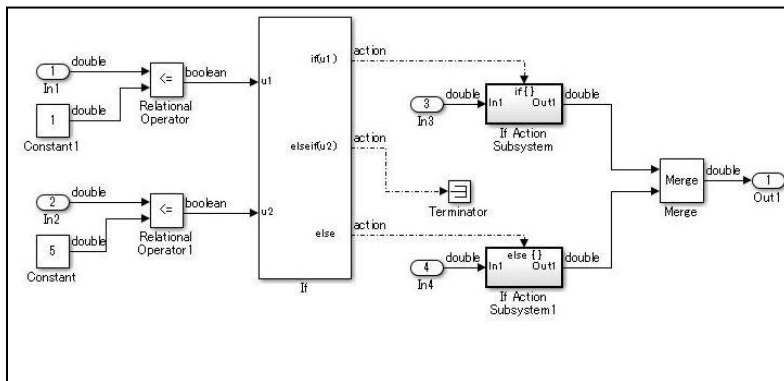
- 意図が異なるルールはIDを分けるべきだが、混乱する可能性があるため、Ver5.0ではサブIDを付与しています。
- 新規ルールについては、意図が違うものはそれぞれIDを付与しています。

- 1つのIDで複数のルールを記述している場合、ルールごとにサブIDを付与することで、異なる重要度の設定やサブID単位での採用/不採用の検討が可能になります。
- jc_0121:加減算ブロック(※)の使用方法(Ver5.0)
 - a. 加減算ブロックの{アイコン形状}は、“四角形”を使用します。ただし、フィードバックループの場合は、{アイコン形状}に“丸型”を使用できません。
 - b. 加減算ブロックの第一入力の符号は“+”とします。ただし、フィードバックループの場合は、第一入力の符号に“-”を使用できます。
 - c. 加減算ブロックの入力数は2つまでとします。
- ※ Ver4.0ではSumブロックとして掲載していましたが、意図としては[Sum]や[Subtract]などの{符号リスト}に“+”や“-”が設定できるブロックになります。Ver5.0では用語集として定義しています。

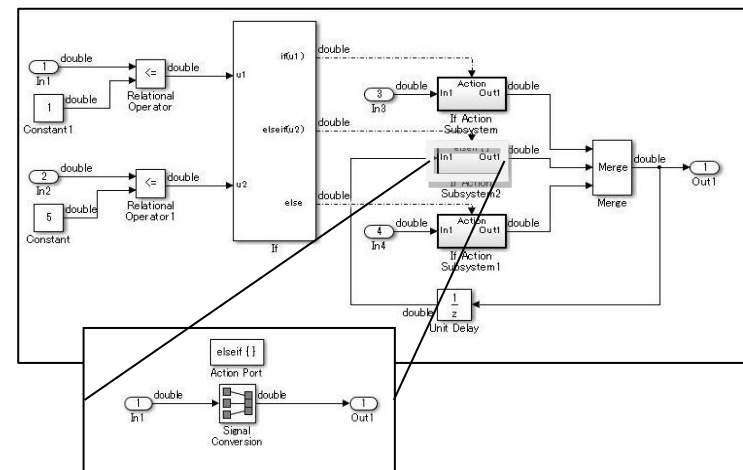
- jc_0657: 条件付制御フローブロックとMergeブロックによる出力値保持

Ver4では、RAM効率が向上する書き方を正
それ以外の記載方法を誤りとしていました。

【正】

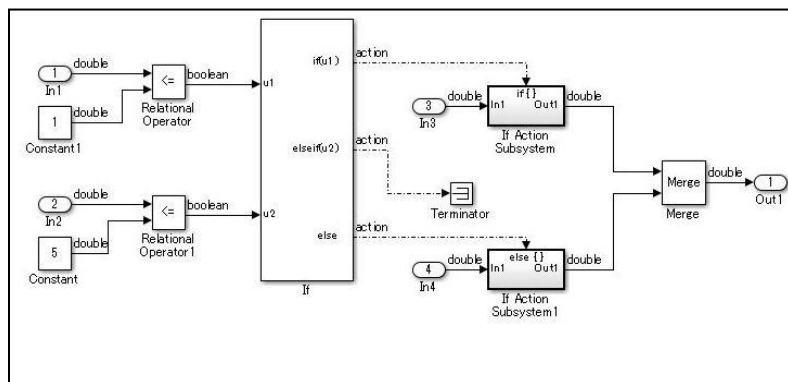


【誤】

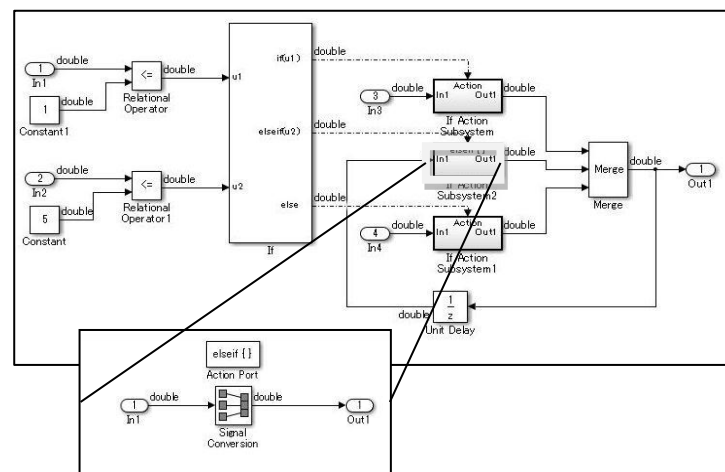


- Ver4.0ではパターン①を【正】、パターン②を【誤】としてルール化していました。
- しかし、ルールの意図は過去値保持のモデリング方法の統一です。Ver5.0では、ユーザーの目的(可読性やコード効率)によって、パターン①,②のどちらかが選択できるように、サブIDのa1,a2として記載しています。

jc_0657a1:



jc_0657a2:



- 同類の異なるルールは、idがa,b,cとして分離する
- 選択式のルールは、idがa1,a2として併記する

系統	末端ID	意味
a系統	a1	選択式のルール
	a2	
	a3	
aとは異なる ルール	b	異なるルール
	c	
	d	

- 1つのIDでいくつのルールが存在するのか、また、それぞれのルールの根拠が紐付けられるよう、Ver5.0ではガイドライン配布時のフォーマットを変更する予定です。

ルールID	jc_0610: 剰余算ブロックの演算子順序		
ルール	サブID	記述内容	重要度
	a	乗除算ブロックの第一入力の符号は"*"とします。	必須
	b	乗除算ブロックの入力数は2つまでとします。	強く推奨
根拠	サブID	記述内容	
	a	浮動小数点の場合、ブロック通りの演算順序((1 ÷ 第一入力) × 第二入力)のコードが生成され、無駄な演算が発生し、期待と異なります。	
	b	演算順序が明確に規定できます。	

ガイドラインの将来

- チェッカを実装後の課題

違反として検出された場合の対処が必要

1. 自動修正

- 簡単なルールほど修正項目が非常に多く自動修正がなければ工数ばかりが必要になり、チェックの意味がなくなる。
- 信頼できるより多くの自動修正が必要。

2. 除外(例db_0032)

- 手動でも、修正不可能な場合は、どうするのか？
- たとえば
 - 交差を0にする。
 - Stateflowの状態数の上限を6にしたが、数値はあくまでも目安であり、修正しないほうが解りやすいケースもある。

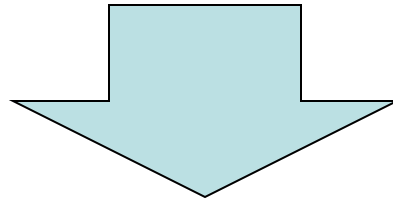
- 外部へ委託後、納品物から見た目に関する違反が500件あった場合、あなたはどうしますか？

1. 全ての違反項目を修正する。
2. ルールをチェック項目から除外する。
3. 無視する。

3番が一番危険です。

そのうち、重要なルールのチェックも意味が無くなります。

- JMAABには、実行結果に影響しない見た目を統一化するルールが多い。
- C言語の、スペースや改行位置の使い方に近いルールが多い。



結果に影響せず、書き方を統一する方法。

1. 最初から規制をかけて、統一化する。
2. 後で、一斉に置換する。

このどちらかで、統一化できる。

- 自動修正は、ランク分けができます。

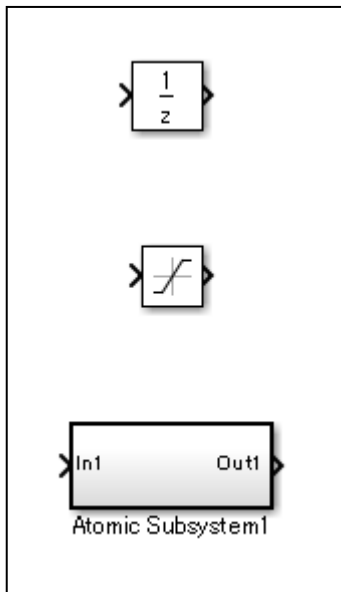
いずれも、前提として実行結果が変わらない事を前提にしています。

	対象ブロック	実行内容
レベル1	1個	チェックボックス等のON
レベル2	1個→1個	ブロックの挿入、削除 ブロックの入れ替え
レベル3	複数→1個	ブロックの挿入、削除 ブロックの入れ替え
レベル4	複数→複数	線の交差を考慮し、ブロック配置までも 綺麗に並べる

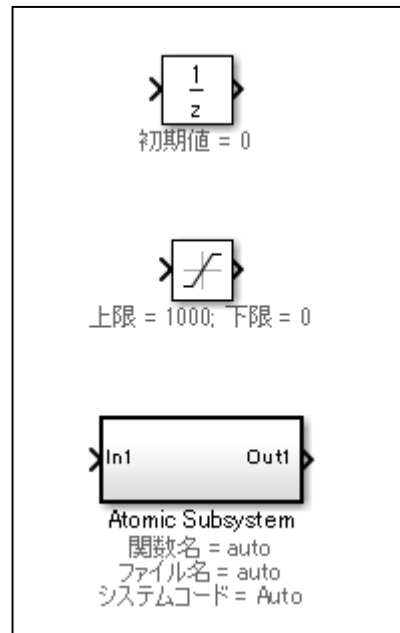
注意:これは、ACC独自のレベル定義です。
JMAABの定義ではありません。

- 決められたアノテーションを表示する

修正前

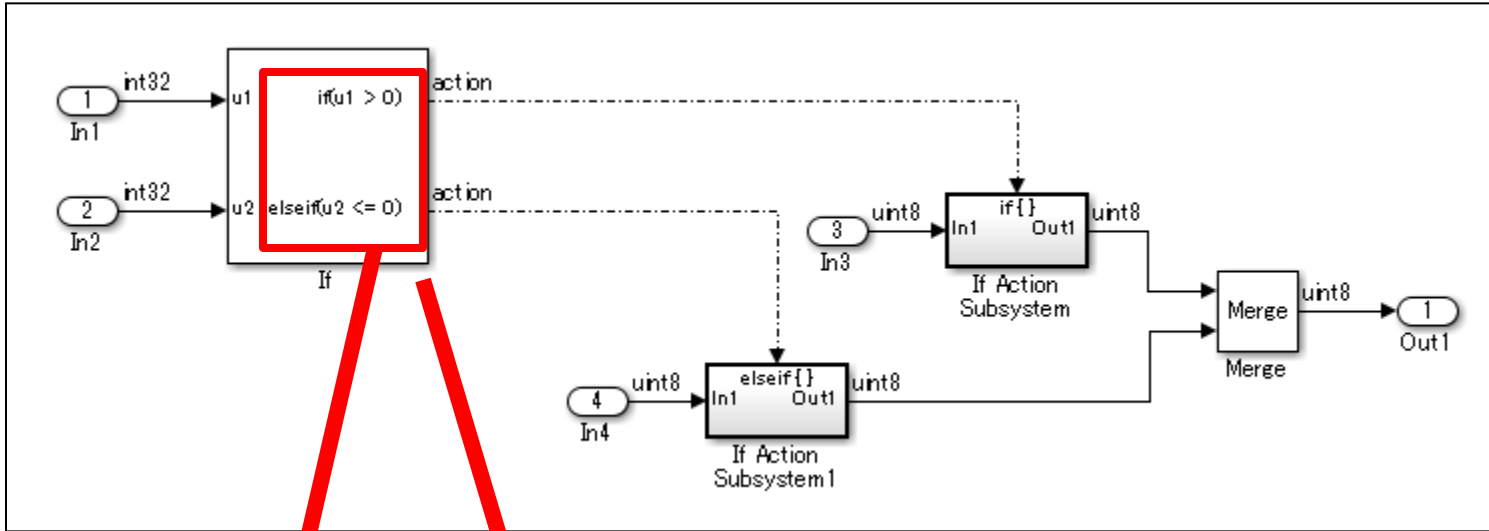


修正後



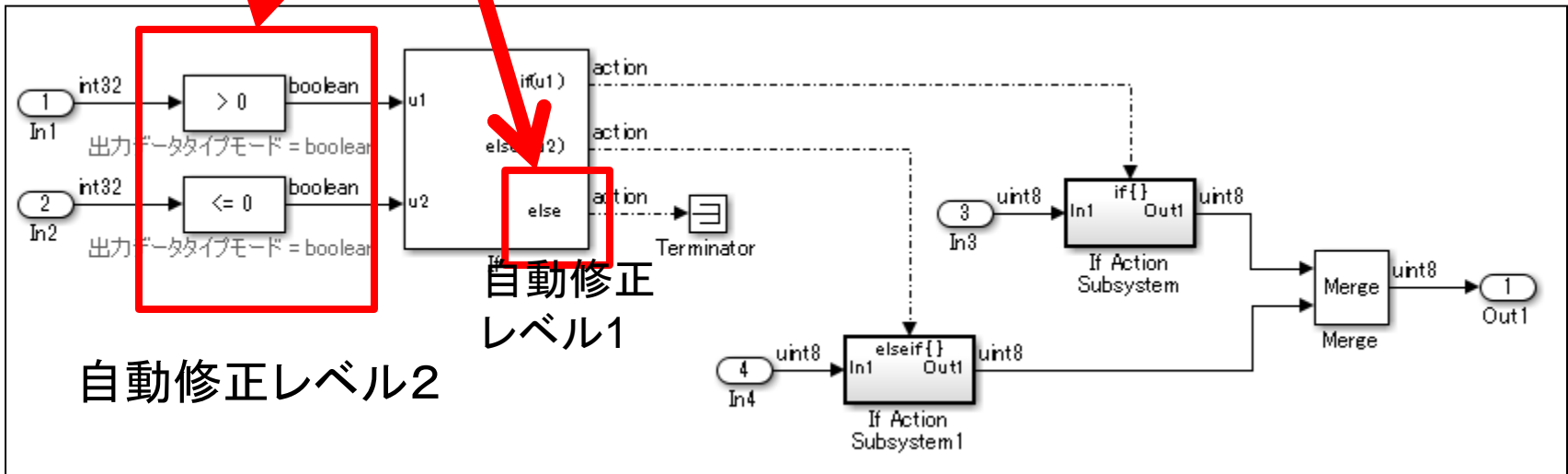
自動修正レベル1

修正前



修正後

elseを追加

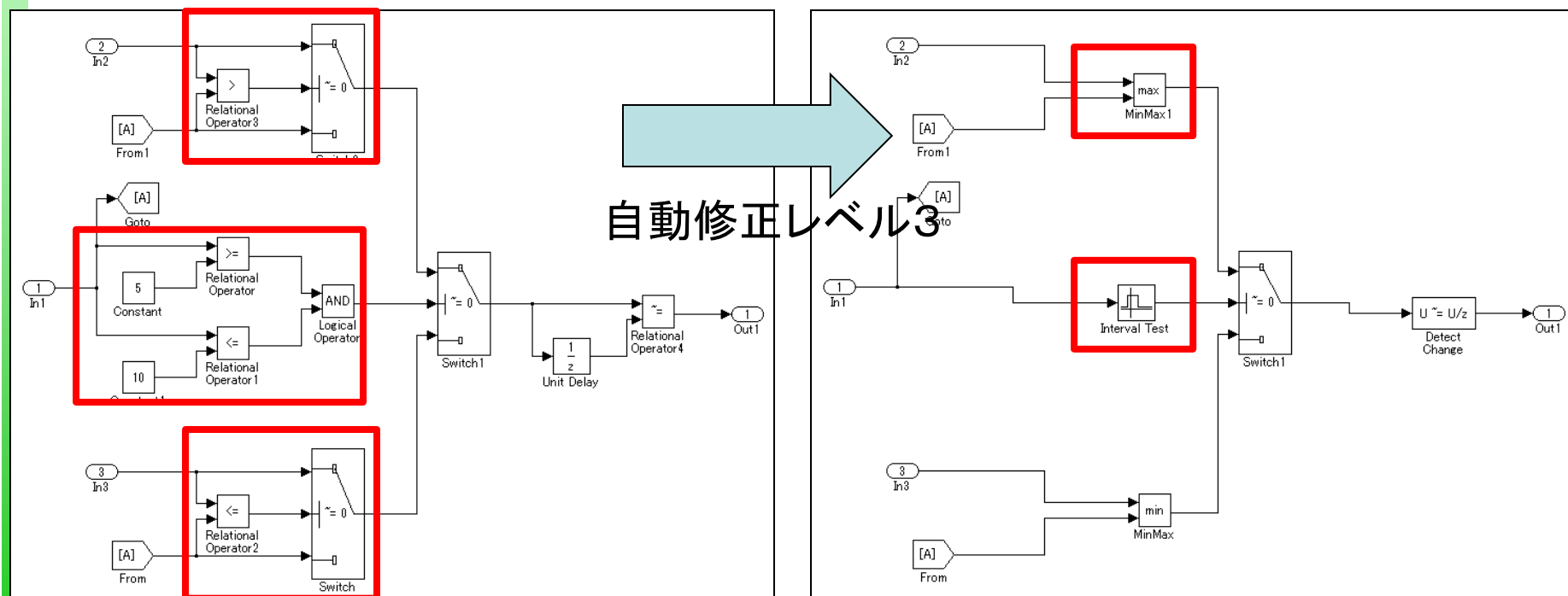


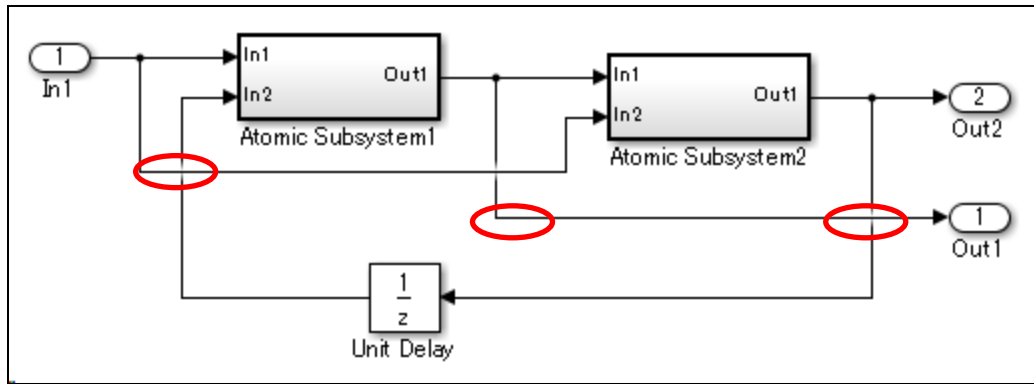
自動修正レベル2

自動修正
レベル1

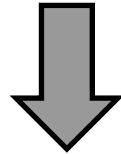
一歩踏み込んだ自動修正

例えば、社内ルール「min,maxは、min,maxブロックを使用する。」
このような例も、自動検出し、自動修正する事が可能です。
使用ブロックの統一は、チェッカーのチェック機能を有効にし、
メンバーの理解向上に役に立ちます。

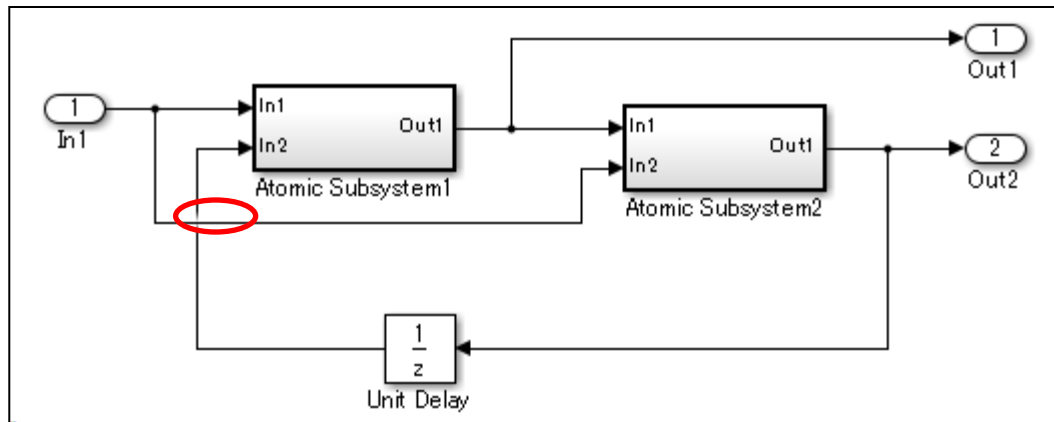




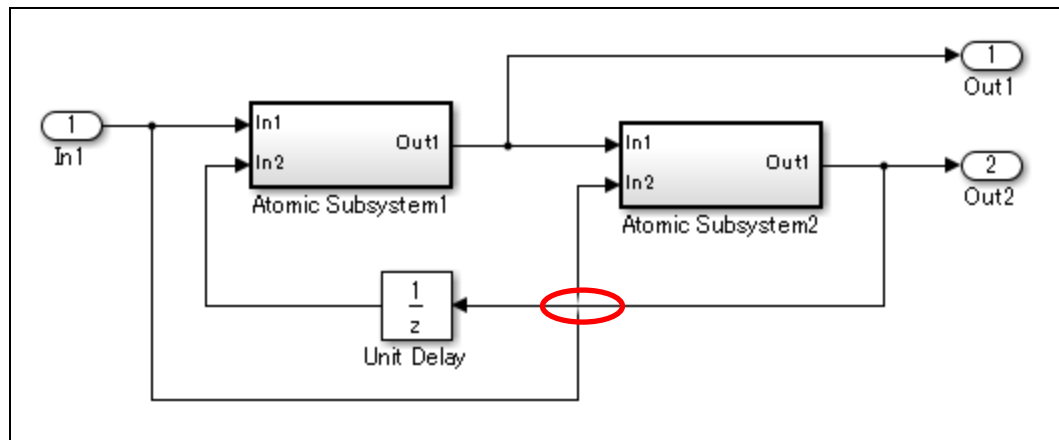
db_0032
線を交差させない。



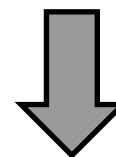
編集しても、交差を0にできない。



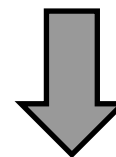
編集しても、交差を0にできない。



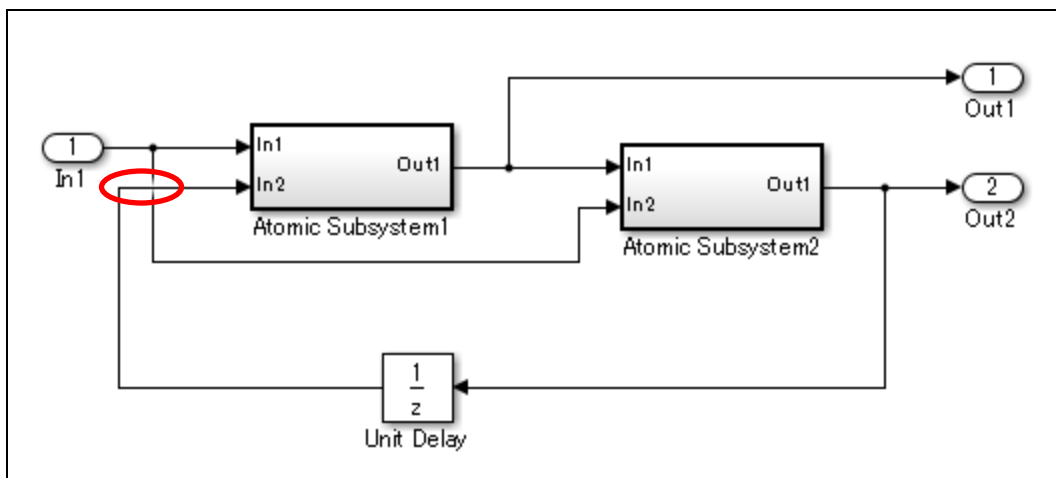
直接結線する場合に、絶対に解決できるかは、不明である。



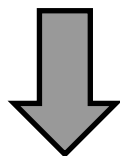
モデルをどう変更しても、Warningが残る場合、どうするのか？



検討して最良になったら pass したい



モデルを修正しても、Warningが残る場合、どうするのか？



モデルに、確認して、交差を最小限にしたマーク
入れて、結果をpassにしたい。

ブロックに対しては、UserDataとして、ルールを確認したという
独自のIDを埋め込み、チェッカーをpassする仕組みが作れる。

PMC05の発表で、紹介します。
興味があれば、聞いてください。

[PMC05] スタイルガイドラインチェッカーの機能の紹介 アイシン・コムクルーズ株式会社 山田 大地 様	[PMD04] ADASアプリのための自律型ドライバモ 築 株式会社デンソー 河合 祐輔 様
[PMC06] FMI適用を想定したツール間の親和性テスト 株式会社デンソー 白井 文祥 様	[PMD05] MBDコンペテ 通したMBDプロセスの アイシン・コムクルーズ株 村田 大輔 様
[PMC07] 並列演算によるシミュレーションの高速化手法 トヨタテクニカルディベロップメント 株式会社 伊藤 忠晴 様	[PMD06] MBDツールチ 有効利用による相乗効 イータス株式会社 青石 勉 様

- ユーザー視点で、使えるガイドラインを目指しています。
 - JMAABスタイルガイドラインはSimulink/Stateflowユーザーの為に作られた、ユーザーの手による、ユーザーの為のガイドライン。
- 良いガイドラインを作ることがJMAABの使命と考え、メンバーで協力しガイドラインの改定を行っています。

Ver5.0は、2017年6月リリース予定です。

ご意見があれば、jmaab-guideline@mathworks.comへ