Vol.2011-SE-171 No.9
2011/3/14

情報制御システム記述言語 RASYS によるモデル記述の網羅性検査ツールの開発

大沼侑司[†1]　小飼敬[†2]　上田賀一[†1]
大久保訓[†3]　高橋勇喜[†3]　中野利彦[†3]

RASYS は情報制御システムを一連の制御判断を基に記述するモデル記述言語である．ルールベースの操作記述によって表現される周期実行処理系では，条件の定義順序や記述網羅性が処理に影響をもたらすため，その事前検査を目的としたツールを検討・開発した．開発したツールを用いて実際の記述モデルの検査を行い，ツールの有効性の確認を行った．

Completeness Checking Tool for RASYS Information Control System Description Model

Yuji Onuma,[†1] Kei Kogai,[†2] Yoshikazu Ueda,[†1]
Satoshi Okubo,[†3] Yuki Takahashi[†3]
and Toshihiko Nakano[†3]

RASYS is a model description language that represents information control systems based on a set of control decisions. In periodic execution system by operation description rule-based, the order and the completeness of conditions influence processing. So, we examined and developed pre-inspection tool. Moreover, to confirm validity of tool, we evaluated the developed tool by using the actual descriptive model.

†1 茨城大学
Ibaraki University
†2 茨城工業高等専門学校
Ibaraki National College of Technology
†3 日立製作所
Hitachi Ltd.

1. はじめに

近年，システムの信頼性や安全性といった品質に対する要求が高まってきている．情報制御システムが社会基盤の多くを担うようになった近年では，システムの不具合が社会へ及ぼす影響は大きなものになってきている．

我々は，品質向上の動向に対応するべく，情報制御システムを対象に専用の情報制御システム記述言語 RASYS を検討・開発してきた[1),2)]．RASYS は情報制御システムの設備や振舞いを，ルールベースに記述していくモデル記述言語である．また，RASYS モデル記述からモデル検査を試みる研究も行っている[3)]．

しかし現在のところ，この RASYS を記述するためのツールや，記述の妥当性を確認するための検査ツールなどは用意されておらず，表計算ソフトで直接モデルの記述を行っている．規模の小さいシステムであれば人手によって正確なモデルを書き上げることも可能であるが，規模が大きくなるにつれて人手によるモデルの正確な記述は難しくなる．RASYS では，モデルの振舞いを制御判断記述に制御条件として記述していく．規模が大きくなれば，条件の組み合わせの数は大幅に増え，条件を漏れなく網羅的に定義するのが困難になる．また，制御条件は記述の上位から順番にチェックされていくのだが，下位の記述で上位の記述に包含されてしまうような条件を記述すると，その条件は意味をなさなくなってしまい最悪の場合，意図した振舞いを表現できていないことになってしまう．

そこで本研究では，これらの問題を解決するために，情報制御システム記述言語を用いてモデルを記述する際に，モデルの記述を補助するためのツールを作成する．本研究では，対象の情報制御システム言語を RASYS とし，そのシステムの振舞いを司る制御判断記述に着目し，モデルの制御条件が網羅的に定義されるかという点に焦点を当てる．このツールを用いることで，制御条件が網羅的にかつ正確に記述することが可能になる．

2. 情報制御システム記述言語 RASYS について

本研究が対象としている情報制御システム記述言語 RASYS について説明する．

2.1 記述言語の概要

今回対象とする RASYS は，情報制御システムを対象として開発されているモデル記述言語である[1)]．システムの振舞いや実行条件はそれぞれ専用の記法によってモデル化され，複数のモデルを組み合わせることでシステムの全体を表現している．

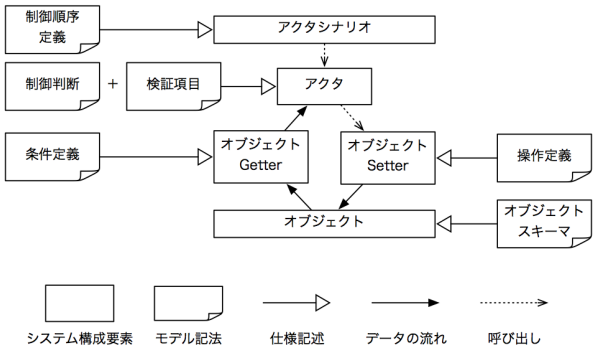

図 1 システムの構成要素とモデル記法の対応

2.2 モデルの構成

モデルは以下の 3 つの要素から構成されている．

- オブジェクト
  システムを構成する設備を表す．設備の状態等の属性とその属性値を持つ．オブジェクトは設備に対応するインスタンスとして実装され，個々の設備とインスタンスは 1 対 1 に対応する．設備の状態は基本的に離散化された値として属性が持つ．
- アクタ
  オブジェクトで定義した設備が持つ属性値の参照や更新などの操作を表す．アクタはシステムの振舞いや制約条件をルールベースのモデルとして表現している．
- アクタシナリオ
  アクタを順次呼び出して実行する関数手続きにて実装する．前のアクタの実行が終了しない限り，次のアクタは実行しない．

2.3 モデルの記法

実行可能な情報制御システムとなるために，6 種類のモデル記法でシステムの構成要素を記述する．その概要を図 1 に示す．

オブジェクト

オブジェクトは以下の 3 種類の記法から構成される．

- 条件定義
  制御条件の定義を表す．制御条件は他の制御条件との依存関係も表現する．制御条件はオブジェクトの属性値を組み合わせて表現される．
- 操作定義
  制御判断において制御条件が成立したときに実行する操作を定義する．操作は，操作の条件と条件成立時に実行する操作内容の組で表す．条件と操作内容の表記については，制御判断と同様である．
- オブジェクトスキーマ
  オブジェクトが持つ属性一覧と他のオブジェクトとの依存関係を表す．

ア ク タ

アクタは以下の 2 種類の記法から構成される．

- 制御判断
  制御条件と，その条件が成立した際に実行する操作の組の集合を表す．制御条件は条件定義で定義されたものを使用し，実行する操作は操作定義で定義される操作になる．
- 検証項目
  制御判断において，検証したいシステムの状態を表す．

アクタシナリオ

アクタシナリオは以下の 1 種類の記法から構成される．

- 制御順序定義
  制御判断が示す制御内容の実行順序を示す．

3. 検査ツール

本研究では，前章までに説明した RASYS を用いた情報制御システム記述モデルの作成支援をするためのツールを開発した．

3.1 ツールの目的

このツールは，制御判断における制御条件の網羅性と有効性を判断するための支援を目的としている．

具体的には以下のような点に注目する．

- 制御判断の制御条件に漏れがなく，網羅的に条件を定義できているか．
  制御判断に記述されている制御条件が，起こり得る条件の組み合わせに対して網羅的に記述されているのか．条件に漏れがある場合，どのような条件に漏れが発生しているのか．
- 各々の制御条件は起こり得る条件として有効に定義されているか．

条件が起こり得る組み合わせに対して定義されているのか．

- 各々の制御条件は有効な順位に定義されているか．
  上位で記述されている条件に包含されてしまうような条件が下位に記述されていないか．その場合，条件の順位をどのように変更したら良いか．

これらの問題を発見，解決できるようなツールを開発することを目的とする．

3.2 ツールの概要

本ツールの概要を図 2 に示す．このツールは Java で開発されている．RASYS モデルの制御判断とオブジェクトスキーマを入力として，制御判断の記述の妥当性の検査を行う．RASYS モデル記述は CSV 形式で記述されたものを入力とする．ツールは，入力ファイル解析，状態生成，状態分類，状態照合の手順で解析を行う．

入力ファイル解析

ツールは RASYS の制御判断とオブジェクトスキーマを入力データとしている．データはツール起動時に選択して入力する．入力されたデータは解析され，データクラスに格納される．

状 態 生 成

ツールは入力された制御判断の中で，制御条件として使われているオブジェクトスキーマを読み込む．次に読み込まれたオブジェクトスキーマの属性値を読み込む．これが終わると，ツールは制御判断の条件とオブジェクトスキーマの属性値から，起こり得るすべての状態の組み合わせを生成する．

状 態 分 類

状態生成で生成された全状態を，制御判断で定義されている制御条件ごとに分類する．この時は制御条件の定義されている順位は考慮せずに，単に各々の制御条件に対して該当する状態を発見する．これ以降，ここで発見された各々の制御条件に該当する状態は，各々の条件に対する状態集合として取り扱われる．

状 態 照 合

状態分類で分類された各制御条件に対する状態集合を，制御条件の定義順位に従い照合していく．照合では，上位の制御条件のので該当した状態集合から下位へと順番に和集合をとり，網羅性の検査を行っていく．この際，一つ上の状態までの和集合と現在の状態の状態集合の積集合をとり，該当する状態の重複の検査も行う．

3.3 ツールの構成

本ツールは，入力データを解析するパーサ部分とデータを格納するデータクラス部分，モデルの検査を行う検査部分から構成される．図 3 に構成を示す．

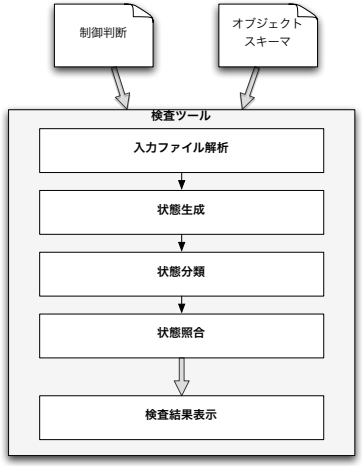

図 2 検査ツールの概要

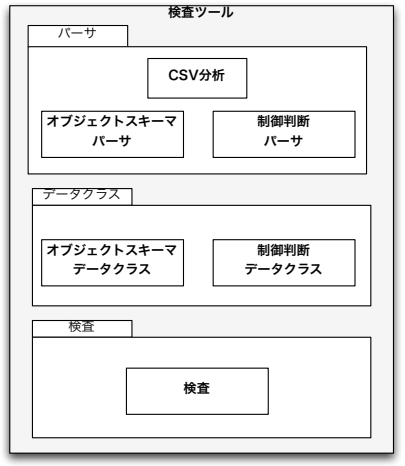

図 3 検査ツールの構成

3.3.1 パーサパッケージ

本ツールの入力データは CSV としている．CSV の読み取りにはオープンソースで開発されている OpenCSV[4] を使用している．入力されたデータはここで解釈され，データクラスに格納される．

3.3.2 データクラスパッケージ

入力されたデータを保持する．今回は，使用するデータの制御条件とオブジェクトスキーマそれぞれのデータクラスを用意した．

3.3.3 検査パッケージ

まず始めに，制御判断が制御条件で参照するオブジェクトスキーマを取得する．取得したオブジェクトスキーマにより，発生し得るすべての状態組み合わせの集合を作成する．制御判断の制御条件を，作成した全状態の集合から適用して各々の条件に該当したものを抽出して状態集合を作る．

次に，先ほど作った各々の条件に該当した状態集合を，制御判断の上位に記述されたものから順番に和集合をとって網羅性を，積集合をとって順位の有効性を検査していく．

なお，状態集合は Java の ArrayList で保持されている．ArrayList の集合演算は Java の

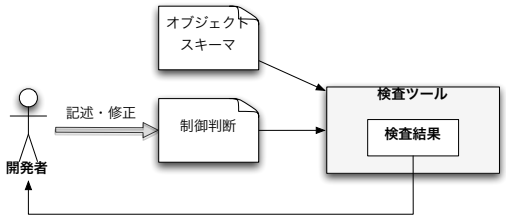

図 4 検査ツールの適用場面

標準ライブラリではサポートされていないため，今回は Apache Commons[5] で提供されている Collections ライブラリを利用した．

3.4 ツールの機能

本ツールでは，次のような機能を実装した．

- 制御条件網羅性検査機能
  条件判断の網羅性を確認するために，どの制御条件にも該当しない状態を探し出す機能．制御判断で記述されている各々の制御条件に該当する状態集合を，全体集合から差集合を取ることによって検査を行う．
- 制御条件有効性確認機能
  各々の制御条件で，意図した状態集合を抽出しているか，条件を定義している順位は適切かを確認する機能．
  ツールで結果を表示する際に，各々の条件に対してその制御条件で該当した状態集合を表示することで前者に対する検査結果を，その制御条件より上位の制御条件で該当した状態集合との積集合を表示することで後者に対する検査結果を提示する．
- 制御条件設定順位入れ換え機能
  制御条件有効性確認機能で条件の順位が適切でなかった場合，順位を入れ替えて再検査を行うことができる機能．

4. 適 用 例

本研究で開発したツールの評価を目的に，実際の RASYS モデルを用いて実験を行った．

4.1 モデル開発の手順

2 章で述べたとおり，RASYS はオブジェクト，アクタ，アクタシナリオから構成されている．現在のところ，モデルは表計算ソフトなどを用いて記述している．通常，システムを構成する設備を表すオブジェクトを定義した後に，設備の振舞いを表すアクタを定義し，最終的にアクタシナリオを定義する．アクタを記述する際にはオブジェクトを参照しながら，アクタシナリオを記述する際にはアクタを参照しながらモデルを構成していく．手順は以下のとおりである．

(1) オブジェクトの記述
- オブジェクトスキーマの記述
- 条件定義の記述
- 操作定義の記述

(2) アクタの記述
- 制御判断の記述
- 検証項目の記述

(3) アクタシナリオの記述
- 制御順序定義の記述

本ツールは，上の手順のアクタの記述の中の制御判断の記述を行う際に用いる．図 4 にツールを適用する場面を示す．

はじめに，開発者はモデルの設備となる 3 種類のオブジェクトスキーマ，条件定義，操作定義のオブジェクトを記述する．

つぎに，記述したオブジェクトをもとに，アクタの記述を行う．モデルの規模により，アクタの制御判断の記述は煩雑になる可能性がある．記述した制御判断は正当性を確認する作業は，さらに煩雑なものとなり得る．ツールは，制御判断の記述の正当性をオブジェクトスキーマと制御判断の 2 つの記述から検査を行う．検査結果は，ツール上で表示されるので，開発者はこれを基に制御判断記述を見直して修正を行う．修正した制御判断は必要に応じて再びツールで検査を行う．この作業を繰り返し行い，制御判断を作成する．

この後，検証項目の記述とアクタシナリオの記述の作成を行う．

4.2 ツールの適用

本研究で開発したツールの評価のために，実際の RASYS モデルの検査を行う．図 5，図 6 に評価に使用した RASYS モデルの制御判断とオブジェクトスキーマを示す．図 7，図 8 がその検査結果画面の一部である．

4.2.1 適用した RASYS モデルについて

モデルは丁字路の信号管理システムをモデル化したものである．丁字路には車用信号機と歩行者用信号機，歩行者用押しボタン，車用センサが設置してある．また，この丁字路は夜

番号	1
モデル種別	制御判断
モデル名称	信号機制御
対象オブジェクト	丁字路

ルール															
No	条件							実行状態							
	6．丁字路 8．通行可能道路変更要求	6．丁字路 9．稼動モード	6．丁字路：4．歩行者用信号機 1．出力	6．丁字路：1．車用信号機A 1．出力	6．丁字路：2．車用信号機B 1．出力	6．丁字路：3．車用信号機C 1．出力	6．丁字路：7．タイマ 2．時間帯	6．丁字路：5．車用センサ 1．反応	6．丁字路：6．歩行者用ボタン 1．押下	6．丁字路：7．タイマ 1．タイマ状態	6．丁字路：4．歩行者用信号機 1．出力	6．丁字路：1．車用信号機A 1．出力	6．丁字路：2．車用信号機B 1．出力	6．丁字路：3．車用信号機C 1．出力	6．丁字路 9．稼動モード
1	-	2．深夜稼動	1．赤	1．赤	1．赤	1．赤	-	-	-	-	4．消灯	-	-	-	-
2	-	2．深夜稼動	4．消灯	1．赤	1．赤	1．赤	-	-	-	-	-	5．赤点滅	-	-	-
3	-	2．深夜稼動	4．消灯	5．赤点滅	1．赤	1．赤	-	-	-	-	-	-	4．黄色点滅	-	-
4	-	2．深夜稼動	4．消灯	5．赤点滅	4．黄色点滅	1．赤	-	-	-	-	-	-	-	4．黄色点滅	-
5	-	2．深夜稼動	4．消灯	5．赤点滅	4．黄色点滅	4．黄色点滅	1．通常	-	-	-	-	-	3．青	-	1．通常稼動
6	1．要求あり	1．通常稼動	1．赤	1．赤	3．青	3．青	-	-	-	-	-	-	-	2．黄	-
7	1．要求あり	1．通常稼動	1．赤	1．赤	3．青	2．黄	-	-	-	-	-	-	-	1．赤	-
8	1．要求あり	1．通常稼動	1．赤	1．赤	3．青	1．赤	-	-	-	-	-	-	2．黄	-	-
9	1．要求あり	1．通常稼動	1．赤	1．赤	2．黄	1．赤	-	-	-	-	-	-	1．赤	-	-
10	1．要求あり	1．通常稼動	1．赤	1．赤	1．赤	1．赤	-	-	-	-	-	3．青	-	-	-
11	1．要求あり	1．通常稼動	1．赤	3．青	1．赤	1．赤	-	1．なし	1．なし	1．ウェイト	3．青	-	-	-	-
12	-	1．通常稼動	3．青	3．青	1．赤	1．赤	-	-	-	-	2．青点滅	-	-	-	-
13	-	1．通常稼動	2．青点滅	3．青	1．赤	1．赤	-	-	-	-	1．赤	-	-	-	-
14	-	1．通常稼動	1．赤	3．青	1．赤	1．赤	-	-	-	-	-	2．黄	-	-	-
15	-	1．通常稼動	1．赤	2．黄	1．赤	1．赤	-	-	-	-	-	1．赤	-	-	-
16	-	1．通常稼動	1．赤	1．赤	1．赤	1．赤	1．通常	-	-	-	-	-	3．青	-	-
17	-	1．通常稼動	1．赤	1．赤	1．赤	1．赤	2．深夜	-	-	-	-	-	-	-	2．深夜稼動
18	-	1．通常稼動	1．赤	1．赤	3．青	1．赤	-	-	-	2．アクティブ	-	-	-	3．青	-

図 5 信号管理システムの RASYS 制御判断記述

間に，車用信号機は点滅信号となり歩行者用信号機は消灯する．

4.2.2 検査結果の考察

ツールの検査結果の考察を，ツールの動作順に行う．

状 態 生 成

図 7 の 1 行目から 6 行目に着目する．

ツールは 3 でも述べたとおり，入力ファイル解析を行った後に，状態生成を行う．状態生成では，制御判断で参照している条件をもとに，起こり得る全状態を生成している．

1 行目を見ると，

“状態組み合わせ：[2, 2, 4, 5, 5, 5, 2]”

という表示がある．ここで図 5 の制御判断記述の制御条件を見る．制御条件は左から

“6．丁字路　8．通行可能道路変更要求”,

“6．丁字路　9．稼動モード”

と順に

“6．丁字路：7．タイマ　2．時間帯”

となっているのがわかる．これらの属性について図 6 のオブジェクトスキーマを参照する．するとそれぞれの属性のもつ属性値の数が 2, 2, 4, 5, 5, 5, 2 個であることがわかり，1 行目の表示と一致する．

これの全状態を生成すると

$$2 \times 2 \times 4 \times 5 \times 5 \times 5 \times 2 = 4000$$

となり，5 行目の 4000 と一致する．

状 態 分 類

次に状態分類について見ていく．

図 7 の 12 行目以降,

第 1 条件:[-1,2,1,1,1,1,-1]:[1,2,1,1,1,1,1][1,2,1,1,1,1,2][2,2,1,1,1,1,1][2,2,1,1,1,1,2]

という表示が各条件に対してある．はじめの

[-1,2,1,1,1,1,-1]

は，制御判断に記述のある制御条件である．ここの “-1” は状態未定義で，どの状態であっても該当させるものである．その後ろにある

[1,2,1,1,1,1,1][1,2,1,1,1,1,2][2,2,1,1,1,1,1][2,2,1,1,1,1,2]

は分類された状態である．

番号	1
モデル種別	オブジェクトスキーマ
モデル名称	車用信号機
実装名称	CarSignal

No	種別	公開	属性	構造		型	データ単位		属性値	初期値
1	状態	○	出力	単一		選択	-	1	赤	○
								2	黄	-
								3	青	-
								4	黄点滅	-
								5	赤点滅	-

番号	2
モデル種別	オブジェクトスキーマ
モデル名称	歩行者用信号機
実装名称	WalkerSignal

No	種別	公開	属性	構造		型	データ単位		属性値	初期値
1	状態	○	出力	単一		選択	-	1	赤	○
								2	青点滅	-
								3	青	-
								4	消灯	-

番号	3
モデル種別	オブジェクトスキーマ
モデル名称	歩行者用ボタン
実装名称	WalkerButton

No	種別	公開	属性	構造		型	データ単位		属性値	初期値
1	状態	○	押下	単一		選択	-	1	なし	○
								2	あり	-

番号	4
モデル種別	オブジェクトスキーマ
モデル名称	車用センサ
実装名称	CarSensor

No	種別	公開	属性	構造		型	データ単位		属性値	初期値
1	状態	○	反応	単一		選択	-	1	なし	○
								2	あり	-

番号	5
モデル種別	オブジェクトスキーマ
モデル名称	タイマ
実装名称	Timer

No	種別	公開	属性	構造		型	データ単位		属性値	初期値
1	状態	○	タイマ状態	単一		選択	-	1	ウェイト	○
								2	アクティブ	-
2	状態	○	時間帯	単一		選択	-	1	通常	○
								2	深夜	-

番号	6
モデル種別	オブジェクトスキーマ
モデル名称	丁字路
実装名称	TShapedRoad

No	種別	公開	属性	構造		型	データ単位		属性値	初期値
1	状態	○	車用信号機A	単一	1	[車用信号機]	-		-	-
2	状態	○	車用信号機B	単一	1	[車用信号機]	-		-	-
3	状態	○	車用信号機C	単一	1	[車用信号機]	-		-	-
4	状態	○	歩行者用信号機A	単一	2	[歩行者用信号機]	-		-	-
5	状態	○	車用センサ	単一	4	[車用センサ]	-		-	-
6	状態	○	歩行者用ボタン	単一	3	[歩行者用ボタン]	-		-	-
7	状態	○	タイマ	単一	5	[タイマ]	-		-	-
8	状態	○	通行可能道路変更要求	単一		選択	-	1	要求あり	-
								2	要求なし	○
9	状態	○	稼動モード	単一		選択	-	1	通常稼動	○
								2	深夜稼動	-

図 6 信号管理システムの RASYS オブジェクトスキーマ記述

状 態 照 合

図 7 の 12 行目以降の

一致 [100, 101, 2100, 2101]

新規 [2100, 2101]

重複 [100, 101]

合計 40

に注目する．これは第 14 条件の 57 行目から 60 行目の表示である．

まず “一致” は，条件分類で一致した状態の通し番号である．すなわち，ここでは第 14 条件で該当した，

[1,1,1,3,1,1,1][1,1,1,3,1,1,2][2,1,1,3,1,1,1][2,1,1,3,1,1,2]

の状態の番号である．

つぎに “新規” と “重複” であるが，“一致” で示された状態のうち，上位の条件で該当しなかった新たな状態が “新規”，上位の条件で既に該当している状態が “重複” で表示されている．

“合計” は，ここまでの条件で該当した状態の合計数である．

解 析 結 果

最終的な解析結果については，図 7 の 64 行目から 69 行目に表示されている．

ここには，制御条件網羅性検査機能の結果が表示されていて，全状態数に対して制御判断に定義された条件で該当した状態件数を表示する．各条件での状態数の推移については，状態照合の合計で表示される．

制御条件有効性確認機能の検査結果については，状態分類と状態照合の部分で表示されている．各々の条件で意図した状態を抽出しているかは状態分類で，条件の定義順位による条件の有効性の確認は状態照合の “新規”，“重複” で確認できる．

制御条件設定順位入れ換え機能

制御条件設定順位入れ換え機能については，図 8 に実行例を示す．

ここでは，図中の 1 行目から 5 行目で，第 11 条件を第 14 条件の後ろに移動させている．その後，もう一度解析を行うと，順位を入れ替えた制御条件での解析結果が表示される．

4.3 ツールの評価

本研究で開発したツールを用いて RASYS モデルを検査することで，制御判断の制御条件の網羅性と有効性の確認を行うことができた．このツールを使うことで従来よりも開発者の負担を軽減することができる．

```
状態組み合わせ : [2, 2, 4, 5, 5, 5, 2]

-----状態生成開始-----
順列生成完了.
状態数 : 4000
時間 : 24[ms]

-----状態分類・照合開始-----
条件分類完了.
時間 : 117[ms]

第 1 条件 : [-1, 2, 1, 1, 1, 1, -1] : [1, 2, 1, 1, 1, 1, 1][1, 2, 1, 1, 1, 1, 2][2, 2, 1, 1, 1, 1, 1][2, 2, 1, 1, 1, 1, 2]
    一致      [1000, 1001, 3000, 3001]
    新規      [1000, 1001, 3000, 3001]
    重複      []
    合計      4

第 2 条件 : [-1, 2, 4, 1, 1, 1, -1] : [1, 2, 4, 1, 1, 1, 1][1, 2, 4, 1, 1, 1, 2][2, 2, 4, 1, 1, 1, 1][2, 2, 4, 1, 1, 1, 2]
    一致      [1750, 1751, 3750, 3751]
    新規      [1750, 1751, 3750, 3751]
    重複      []
    合計      8

第 3 条件 : [-1, 2, 4, 5, 1, 1, -1] : [1, 2, 4, 5, 1, 1, 1][1, 2, 4, 5, 1, 1, 2][2, 2, 4, 5, 1, 1, 1][2, 2, 4, 5, 1, 1, 2]
    一致      [1950, 1951, 3950, 3951]
    新規      [1950, 1951, 3950, 3951]
    重複      []
    合計      12

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~省略~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

第 10 条件 : [1, 1, 1, 1, 1, 1, -1] : [1, 1, 1, 1, 1, 1, 1][1, 1, 1, 1, 1, 1, 2]
    一致      [0, 1]
    新規      [0, 1]
    重複      []
    合計      28

第 11 条件 : [1, 1, 1, 3, 1, 1, -1] : [1, 1, 1, 3, 1, 1, 1][1, 1, 1, 3, 1, 1, 2]
    一致      [100, 101]
    新規      [100, 101]
    重複      []
    合計      30

第 12 条件 : [-1, 1, 3, 3, 1, 1, -1] : [1, 1, 3, 3, 1, 1, 1][1, 1, 3, 3, 1, 1, 2][2, 1, 3, 3, 1, 1, 1][2, 1, 3, 3, 1, 1, 2]
    一致      [600, 601, 2600, 2601]
    新規      [600, 601, 2600, 2601]
    重複      []
    合計      34

第 13 条件 : [-1, 1, 2, 3, 1, 1, -1] : [1, 1, 2, 3, 1, 1, 1][1, 1, 2, 3, 1, 1, 2][2, 1, 2, 3, 1, 1, 1][2, 1, 2, 3, 1, 1, 2]
    一致      [350, 351, 2350, 2351]
    新規      [350, 351, 2350, 2351]
    重複      []
    合計      38

第 14 条件 : [-1, 1, 1, 3, 1, 1, -1] : [1, 1, 1, 3, 1, 1, 1][1, 1, 1, 3, 1, 1, 2][2, 1, 1, 3, 1, 1, 1][2, 1, 1, 3, 1, 1, 2]
    一致      [100, 101, 2100, 2101]
    新規      [2100, 2101]
    重複      [100, 101]
    合計      40

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~省略~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

-----解析結果-----
全状態数        : 4000
検査除外状態数  : 0
検査対象状態数  : 4000
検査状態数      : 48
未検査状態数    : 3962
command:exit
```

図 7 信号管理システム検査結果画面（一部抜粋）

```
command:edit
条件の番号:
11
挿入先:
14
command:analyze

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~省略~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

第 11 条件 : [-1, 1, 3, 3, 1, 1, -1] : [1, 1, 3, 3, 1, 1, 1][1, 1, 3, 3, 1, 1, 2][2, 1, 3, 3, 1, 1, 1][2, 1, 3, 3, 1, 1, 2]
    一致      [600, 601, 2600, 2601]
    新規      [600, 601, 2600, 2601]
    重複      []
    合計      32

第 12 条件 : [-1, 1, 2, 3, 1, 1, -1] : [1, 1, 2, 3, 1, 1, 1][1, 1, 2, 3, 1, 1, 2][2, 1, 2, 3, 1, 1, 1][2, 1, 2, 3, 1, 1, 2]
    一致      [350, 351, 2350, 2351]
    新規      [350, 351, 2350, 2351]
    重複      []
    合計      36

第 13 条件 : [-1, 1, 1, 3, 1, 1, -1] : [1, 1, 1, 3, 1, 1, 1][1, 1, 1, 3, 1, 1, 2][2, 1, 1, 3, 1, 1, 1][2, 1, 1, 3, 1, 1, 2]
    一致      [100, 101, 2100, 2101]
    新規      [100, 101, 2100, 2101]
    重複      []
    合計      40

第 14 条件 : [1, 1, 1, 3, 1, 1, -1] : [1, 1, 1, 3, 1, 1, 1][1, 1, 1, 3, 1, 1, 2]
    一致      [100, 101]
    新規      []
    重複      [100, 101]
    合計      40
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~省略~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

図 8 信号管理システム制御条件設定順位入れ換え機能画面（一部抜粋）

しかし，ツールがコマンドラインベースであったり，検査結果の表示がわかりづらいなど，改善の余地がある．

また，本論文の評価では行っていないが予備実験の際，状態数が膨大になるような制御判断の RASYS モデルを本ツールで検査すると，ツールがエラー停止してしまった．原因は状態生成の段階で，メモリ使用量がヒープメモリサイズを超えてしまっていることで，現状では全状態数が 500 万状態程度までのモデルであればツールの正常動作確認が取れている．

5. おわりに

5.1 まとめ

本研究では，情報制御システム記述言語 RASYS を対象に制御判断記述の網羅性を検査するツールを開発した．これにより，従来モデル記述より手作業で確認していた制御条件の網羅性と定義する順位による有効性を自動的に得ることができる．本ツールを利用することで，RASYS モデルの作成・修正の手間を減らし，より正確かつ速やかに情報制御システム

モデルを記述することができる．

本研究で作成したツールは，ソフトウェアのテストの段階でテストパターンを網羅的に生成する[6] ものとは異なり，モデルを作成する段階でのモデル記述の網羅性検査を行うものである．また，網羅的検証を行うモデル検査ツール[7][8] もあるが，これらはシステムとは別に検証すべきモデルを作成しなければならない点で，本研究のツールとは異なる．

5.2 今後の課題

本研究の成果から，今後の研究課題として以下のようなものが存在すると考えている．

- 状態数が増えた場合への対応
  今回のツールの実装方法では状態数が増えたときにツールが停止してしまう．規模の大きなシステムのモデルを作成する際には，現状のツールでは対応しきれなくなってしまう．状態数が増えた場合でもツールが正しく動作するよう，ツールの実装方法や開発言語などを再検討する必要がある．
- ツールのユーザビリティの向上
  今回開発したツールはコマンドラインベースである．また，結果の表示がユーザにわかりやすい形にはなっていない．ツール上で行った内容の保存等の出力もサポートしていない．ツールを実際の開発現場などで使用することを考えた場合，ユーザの使い勝手というのはとても重要な問題であり，今後の課題としたい．
- RASYS モデル記述スイートの開発
  本研究で開発したツールは，RASYS モデルの制御判断の検査しか行っていない．入力データの CSV に対しても，拡張部分の手直しが必要であったり，想定するモデル記述フォーマットの自由度が低くなっている．
  これらの問題を解決するためには，RASYS モデルの記述を行うためのツールが必要であると考える．RASYS モデルの記述と，今回開発したような記述の検査を行う機能を併せ持つ，モデル記述スイートを開発していく必要があると感じた．

参 考 文 献

1) 小飼敬，上田賀一，大久保訓，高橋勇喜，中野利彦：Alloy を利用した情報制御システム記述言語の仕様検証の実用化，ソフトウェア工学の基礎 XVI，pp.259—266，近代科学社 (2009).
2) 小飼敬，柳翔太，上田賀一，大久保訓，高橋勇喜，中野利彦：検証項目を持つ情報制御システム記述言語のための分析・設計手法，ソフトウェア工学の基礎 XVII，pp.101-106 (2010).
3) 柳翔太，小飼敬，上田賀一，大久保訓，高橋勇喜，中野利彦：情報制御システム記述モデルの検証用記述への変換と効率的検証，日本ソフトウェア科学会第 27 回大会講演論文集，pp.1-9 (2010).
4) opencsv，http://opencsv.sourceforge.net/
5) Apache Commons，http://commons.apache.org/
6) Glenford J. Myers 著，長尾真，松尾正信 訳：ソフトウェア・テストの技法 第 2 版，近代科学社 (2006).
7) Holzmann，G.J.：The Spin Model Checker：Primer and Reference Manual，Addison-Wesley(2004).
8) Daniel Jackson：Alloy: A Lightweight Object Modelling Notation，ACM Transactions on Software Engineering and Methodology，Vol.11，Issue 2，pp.256-290 (2002).