

表計算ソフトによるモデル検査の試行

Trial of model checking by spreadsheet

三菱電機システムサービス株式会社

MITSUBISHI ELECTRIC SYSTEM&SERVICE CO.,LTD.

○野村卓司

○NomuraTakuji

Abstract : Correspondence to the bug that originates in timing is difficult. Therefore, the model checking is paid to attention. However, it is difficult to execute the model checking according to the man-hour etc. of acquisition and the description of the model description language.

We contemplated the execution of the model checking, and executed the improvement of the development process. First of all, it explains the improvement of the executed development process. Next, the tool by the spreadsheet is described. Finally, the trial on business and the improvement in the future are described.

1. はじめに

タイミングが原因となっているバグが発生した場合、対策には多くの工数がかかる。バグを防ぐ手法として、設計段階で検証を行えるモデル検査が注目されている。

しかし、現場で手軽に実施するには、文献が少ない、モデル検査言語記述のコストの問題があった。文献が少ない点については、昨年、書籍が2冊発行された。^(文献[1][2])モデル検査言語の記述コストの削減については、UMLまたは状態遷移表からのモデル検査言語の生成の事例およびツールがある。^(文献[3][4][5])しかし、ツールの導入は、当社の製品は状態遷移の数が小さいことから、コストの点から制約を受ける。

本論文では、普及している表計算ソフトにより作成したツールの紹介およびツールによるモデル検査の試行について報告する。

2. モデル検査試行開始までの改善活動

2006年9月から下記の改善を行った。

BCEアーキテクチャの採用

(1) プロセスをVモデルからWモデルへ変更

(3) クラス図、表計算ソフトによる状態遷移表の記述

(4) コントロール層の公開メソッドに対する2因子間テストの徹底

その結果、「出荷に際してのテストを、バグ出しからバグのないことを信頼度成長曲線で確認するプロセス」としたことを報告した。^(文献[6])

上記の改善の過程においてモデル検査の実施を視野においていた。プロセスのWモデルへの変更、計算ソフトによる状態遷移表の記述が該当する。モデル検査の実施は、状態遷移表の規模から、Wモデルにおいて、品質保証の担当者が試験項目として作成し、実施することを想定していた。

状態遷移の記述をUMLのステートマシン図にせず、状態遷移表にしたのは、以下の理由である。

- (1) 記述の漏れがチェックし易い
- (2) 表計算ソフトがあれば記述可能
- (3) マクロで状態遷移表からスケルトン、モデル検査言語が生成できる

三菱電機システムサービス株式会社 産業システムセンター システムエンジニアリング部
Mitsubishi Electric System&Service Co. Industrial system center System engineering part

〒461-8675 名古屋市東区矢田南5丁目1番14号 TEL 052-722-8711

5-1-14 Yataminami, Higasi, Nagoya, Japan TEL 052-722-8711

上記の改善と併行して、表計算ソフトによるモデル検査の実施例およびツールの動向を注視していたが、該当するものは無かった。改善の結果がまとまった為、新たな改善策の一つとして、信頼性向上を目的に、ツールを内作り、モデル検査の試行を行った。^(文献[6])

3. 作成したツール

3.1 ツールの概要

外部仕様は、表計算ソフトの1シートに対し、状態遷移表を1つ記述した。以下の2つの機能を実現した。(1)のみのツールを作成した試行を開始したが、反例解析の効率改善のために(2)の機能を追加した。

- (1) 状態遷移表を記述しているシート名をプロセス名としてPromela^(注[1])を生成する。
- (2) (1)で作成した状態遷移の履歴を反例発生前の指定した件数読み込み、表形式で表示する。表を基に、シートのフォーカスの移動およびセルの色の変更によって、反例発生前の状態遷移を再現する。

上記の機能は、以下のようにして実現した。

- (1) 図1に示す状態遷移表を図3に示すデータ構造への変換後、Promelaに変換した。
状態遷移表の表現の変換機能および状態遷移図からの変換機能の追加を考慮して、一旦、図3データ構造に変換した後、Promelaに変換した。
- (2) モデル検査の検証の過程における状態遷移は、図2の構造のデータが、図4のように実行順に並んだものである。状態遷移の履歴が trail.out^(注[2])ファイルに記録されるようにprintf文を生成した。

3.2 データ構造

現在使用している状態遷移表の書式を図1に示す。

縦軸に状態、横軸にイベントを列挙する。状態遷移は、条件、アクション、遷移先状態を3つのセルで記述する。

	イベント1	イベント2	・	・	イベントN
状態1	条件				
	アクション	←一つの状態遷移を3つのセルで記述			
	遷移先状態				
状態2					
⋮					
状態M					

図1 状態遷移表の書式

プロセス名	現在の状態	イベント	条件	アクション	遷移先状態
-------	-------	------	----	-------	-------

図2 基本的なデータ構造

プロセス1	状態1	イベント1	条件1	アクション	遷移先状態
プロセス1	状態1	イベント2	条件1		遷移先状態
		・			
プロセスN	状態M	イベント1	条件1		遷移先状態

図3 Promela生成に適するようにソート

プロセスA	現在の状態	イベント	条件	アクション	遷移先状態
プロセスB	現在の状態	イベント	条件	アクション	遷移先状態
		・			
プロセスA	現在の状態	イベント	条件	アクション	遷移先状態
プロセスZ	現在の状態	イベント	条件	アクション	遷移先状態

図4 検査結果のデータの並び

注[1] 検査ためにモデルを記述する言語。

注[2] 反例が出力されるファイル。反例とは、検査しようとする仕様が満たされない検証結果。

3.3 生成した Promela の概要

図3のデータを以下の構造の Promela に変換した。(文献[1][2])

状態遷移表から宣言を作成

active proctype プロセス名() ←シート名をプロセス名とした
{状態遷移表から宣言を作成

do

:: イベントがない遷移; printf(図2のデータ構造)

:: (イベント1)→

if 状態

条件ごとに、アクション、遷移先の状態、printf(図2のデータ構造)を記述

例) 現在の状態が sts1 で

(a == 1), ch!msg, sts=sts2, printf(“keyword, sts1, (a==1), ch!msg, sts2”)

...

fi;

:: (イベント2)→

:: (イベントn)→

od}

参考にした文献において、テンプレートとして2つ記述されているが、下記の点を考慮して上記のパターンを採用した。(文献[1])

- (1) モデル検査の導入を困難にすることに、検査式の作成がある。検査式に容易に状態遷移を表現できる為、状態を変数として持つパターンを採用した。(文献[2])
- (2) 実装する言語との整合性。

3.4 自動で生成する宣言

状態遷移表の記述内容を基に、Promela の作成効率改善の為に下記の宣言を作成した。

- (1) チャンネルの通信に使用されている変数は、チャンネルの定義を参照して宣言。
- (2) 状態遷移表の条件に使用されている変数をデフォルト mtype として宣言。
- (3) アクションとして出てくる名前を、ダミーの変数を初期化する処理のみを inline として定義。

4. 業務での試行

4.1 プロセスにおけるモデル検査の位置づけ

対象となるシステムは、お客様からのオーダーに対応するオーダー品と自社で企画する開発品がある。開発品の方が、開発規模が大きく、複雑な状態遷移をとる。

現在、設計者が記述した状態遷移表を品質保証の担当者が、Promela に変換して、モデル検査を試行している。

4.2 試行結果について

- (1) オーダー品における試行

開発プロセスはVモデルである。状態遷移が小規模である為、品質保証担当者が仕様の理解を深める為にシミュレーションを行った。

- (2) 開発品における試行

製品毎に品質保証と設計でチームを編成する。次頁図5に示すように、試験仕様書作成の一環として、モデルの作成を試行した。

開発品は、オーダー品より状態遷移および併行して動作するプロセスの数が多いため、検証を行った。3.1で述べたように反例発生前の状態遷移の履歴を、表計算ソフトのシートのフォーカスの移動とセルの色の変化で表現できる。その為、解析は効率をあげる為に、打ち合わせおよびデザインレビューにおいて、設計者と共同で行なった。

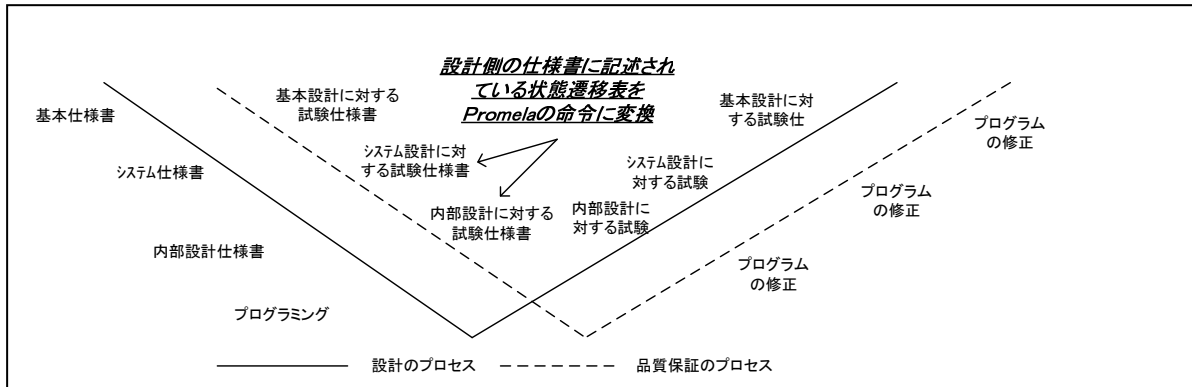


図5 モデル検査の実施

サーバの2重化に対し、モデル検査を実施した。

各プロセス状態を持つための変数、イベント処理用の変数を広域変数としていた為、状態爆発を起こした。各プロセスの状態を持つための変数を局所変数として Promela を生成し、必要に応じて広域変数にした。

5. ツールを内作したメリット

検証結果に対し、設計から、反例が発生する状況を説明してほしいと指摘された。この点に対し、3.1で述べたように、反例発生前の解析機能を追加した。これにより、検証結果を検討してもらえるようになった。今後も6.で述べる改良を検討しているが、図2に示したデータ構造を決め、ツールを内作したメリットであると思う。

6. 試行からプロセスへの定着にむけて

試行を通じ、モデル検査は、品質保証担当にとり、重要なスキルになると感じた。下記の課題に取り組むとともに、検証およびテスト設計^(文献[7])に作成したツールを活用してゆきたい。

(1) 変換機能の改善

- ①設計者の作成する状態遷移表の表現を Promela に変換する機能
- ②宣言洗い出し機能の強化

(2) 反例の為に出力したデータを解析し易い表現にする

(3) モデル検査の実施の為に書式の作成

設計者の状態遷移表をモデル検査の為に Promela の表現に変換する際、設計者と品質保証担当のコミュニケーションをとる為にモデル記述用の書式を決めたい。

[謝辞]

今回の試行にあたり、森島誠、松本豊両氏の多大な協力に感謝する。

[参考文献]

[1] 中島 震、「SPIN モデル検査」、2008 年

[2] 萩谷 昌巳、吉岡 信和、青木 利晃、田原 康之「SPIN による設計モデル検証」、2008 年

[3] 村石理恵/服部彰宏/野村秀樹/山本 訓稔「ModelChecking を適用した実践的非同期制御検証」
<http://www.jasst.jp/archives/jasst07e/pdf/D2-3.pdf>、2007 年

[4] 小池隆、矢野恭、<http://unit.aist.go.jp/cvs/consortium/cons081205/cons081205.html>、2008 年

[5] <http://www.zipc.com/support/rre/spinconv/>

[6] 野村 卓司/松本 豊/本間 幹和、「テストによらない品質保証を目指して」
<http://www.jasst.jp/archives/jasst09e/pdf/B5-1.pdf>、2009 年

[7]<http://ja.wikipedia.org/wiki/%E3%83%A2%E3%83%87%E3%83%AB%E3%83%99%E3%83%BC%E3%82%B9%E3%83%86%E3%82%B9%E3%83%88>