

## AUTOSAR OS に対する テストケースおよびテストプログラムの自動生成 —RTOSへのPictMasterの適用—

風間 佳之<sup>1)</sup> 平橋 航<sup>2)</sup> 鶴原 一人<sup>2)</sup> 海上 智昭<sup>2)</sup> 本田 晋也<sup>3)</sup> 高田 広章<sup>3)</sup>

1) 日本電気通信システム株式会社 組込みシステム事業本部 〒448-0000 愛知県刈谷市東日成 28-1

2),3) 名古屋大学 大学院情報科学研究科 〒464-8601 愛知県名古屋市千種区不老町

E-mail: 1) kazama.ys@ncos.nec.co.jp, 2) {whirahashi, shigihara, unagami}@nces.is.nagoya-u.ac.jp,

3) {honda, hiro}@ertl.jp

**あらまし** 本論文では、PictMaster を用いた AUTOSAR OS 仕様ベースの RTOS に対する テストケースおよび テストプログラムの自動生成に関する報告する。名古屋大学の組込みシステム研究センターにて実施した AUTOSAR OS 仕様ベースの RTOS 及び テストスイート の開発では、AUTOSAR OS 仕様におけるタスクの種別やスケジューリングポリシなどの組み合わせによって、テストケースが膨大となるために手動で テストケース を作成することが困難である。本研究では、組み合わせ テストケース生成ツールである PictMaster を用いて、 テストケース を自動生成することを目的とした。さらに PictMaster を拡張した上で、独自に開発したツールと組み合わせることで、 テストケース から テストプログラム の自動生成までを行った。また、PictMaster を用いた テストケース および テストプログラム の自動生成に関する有用性および、今後の課題について論じた。

**キーワード** 組込みリアルタイム OS, AUTOSAR OS, TOPPERS, オープンソース, システムサービステスト, テストツール

## Automatic Generation of Test Case and Test Program for AUTOSAR OS —Implementation of PictMaster to RTOS—

Yoshiyuki KAZAMA<sup>1)</sup> Wataru HIRAHASHI<sup>2)</sup> Kazuto SHIGIHARA<sup>2)</sup> Tomoaki UNAGAMI<sup>2)</sup> Shinya HONDA<sup>3)</sup>  
and Hiroaki TAKADA<sup>3)</sup>

E-mail: 1) kazama.ys@ncos.nec.co.jp, 2) {whirahashi, shigihara, unagami}@nces.is.nagoya-u.ac.jp,

3) {honda, hiro}@ertl.jp

1) NEC Communication Systems, Ltd. Embedded Systems Operations Unit  
28-1 Higashihinari, Kariya-shi, Aichi, 448-0000 Japan

2),3) Graduate School of Information Science, Nagoya University Furocho, Chikusa-ku, Nagoya-shi, Aichi, 464-8601 Japan

**Abstract** The present paper reports automatic generation of test case and test program by PictMaster for RTOS based on AUTOSAR OS specification. In the previous research on the development of RTOS based on AUTOSAR OS specification and test suite at the Center for Embedded Computing Systems of Nagoya University, it was revealed that the combination of several task types and scheduling policy in AUTOSAR OS specification results in a vast increase in test case which causes hardships in manual generation of test case. In the present paper, the authors adopted combination testing tool PictMaster to generate test case automatically. The authors enhanced the function of PictMaster and combined it with an originally developed tool to automatically generate test program from test case. Serviceability of automated generation of test case and test program by PictMaster were discussed, along with some future directions.

**Keyword** Embedded real time OS, AUTOSAR OS, TOPPERS, Open source, System service test, Test tool

## 1. はじめに

近年の車載システムには、省エネルギー化や安全性向上をねらい、高度かつ複雑な電子制御が求められるようになっている。また、複雑化した ECU を統合し、シンプルな構成にすることで、新しい機能やサービスを提供することが重要な課題と位置づけられている。これらの課題を解決するために、車載ソフトウェアの標準化を行うコンソーシアムである AUTOSAR [1]から、OSEK/VDX OS[2]仕様の上位互換に相当する、AUTOSAR OS の仕様が開示された。これらの OS は、車載システムで利用されることを前提としているため、実装して利用する際には高い品質が求められる。品質確保のためには十分なテストが必要であるが、AUTOSAR OS はメモリ保護やタイミング保護、マルチコアなどの機能を有しているため、テストには一定のコストを要すると考えられる。

名古屋大学大学院情報科学研究科附属組込みシステム研究センター(NCES) [3]は、コンソーシアム型の研究組織を立ち上げ、複数の企業の参加を得て、AUTOSAR OS 仕様に対応する次世代 RTOS の仕様策定を行っている。さらに、策定した仕様に基づいた AUTOSAR OS の実装と、テストスイートの開発を行っている。次世代 RTOS の開発終了後には、TOPPERS プロジェクトより、TOPPERS/ ATK2 (AuTomotive Kernel version2; 以降、ATK2 と略す) の名称で、テストスイートを除き、オープンソースソフトウェアとして公開する計画である。また、策定した RTOS の仕様は、「次世代車載システム向け RTOS 外部仕様書」(以降、外部仕様書と略す)として公開する計画である。

AUTOSAR OS 仕様ベースの RTOS およびテストスイートの開発では、AUTOSAR OS 仕様におけるタスクの種別やスケジューリングポリシなどの組み合わせによって、テストケースが膨大となる。さらに、メモリ保護やタイミング保護も組み合わせた場合、手動でテストケースを生成することが困難であることが予想される。したがって、テストケースの自動生成は、AUTOSAR 仕様 OS ベースの RTOS およびテストスイート開発において極めて現実的な課題のひとつとしてあげることができる。本研究では、組み合わせテストケース生成ツールである PictMaster を用いて、テストケースを自動生成することを目的とした。また、PictMaster を用いたテストケースおよびテストプログラムの自動生成の有用性および、今後の課題について論じた。

本論文では、前述の研究組織で実施した、ATK2 のシステムサービステスト(以降、SS テストと略す)の開発事例における PictMaster を用いた AUTOSAR OS 仕様ベースの RTOS に対するテストケースおよびテストプ

ログラムの自動生成に関して述べる。

本論文の構成は次のとおりである。まず 2 章において、本研究の実施体制の概要として、コンソーシアム型研究組織を説明する。次いで 3 章では、既存のテストスイートの概要についてまとめ、4 章では本研究プロジェクトのベースとなったテストスイートに起因する問題についてまとめる。さらに、5 章では組み合わせツールに必要な要件についてまとめ、6 章では本研究で用いたツールの選定理由をまとめ。さらに 7 章では実施結果および評価についてまとめ、8 章で本研究成果の今後の展開可能性や課題について、ATK2 開発を中心に据えてまとめ、本研究を総括する。

## 2. コンソーシアム型共同研究

コンソーシアム型共同研究は、NCES 側で設定した研究テーマに対して複数の企業が参加し、NCES を中心としたコンソーシアムで共同研究・開発を行う。研究・開発の成果は、参加企業は自由に使用可能とし、一定期間後にはオープン化する(テストスイートを除く)。なお、本研究組織では、参加する研究者を高度な研究開発型人材として育成することも、目的とする。参加企業の一覧を五十音順で以下に示す。

- 株式会社ヴィッツ
- 株式会社 OTSL
- 株式会社サニー技研
- 株式会社デンソー
- 株式会社東芝
- トヨタ自動車株式会社
- 日本電気通信システム株式会社
- パナソニック アドバンストテクノロジー株式会社
- 富士ソフト株式会社
- 富士通 VLSI 株式会社
- ルネサスエレクトロニクス株式会社

2011 年度は、外部仕様書を作成し、外部仕様書に基づいた実装と、テストスイートの開発を進めている。AUTOSAR OS 仕様では、スケーラビリティクラス(以降、SC と略す)と呼ばれるユーザに必要な機能とターゲットの持つ機能によって OS 機能を定義した 4 段階の機能セット(SC1～SC4)を定義されている。ATK2 開発の第一段階の目標は、SC1 の実装である。

RTOS の検証は、各企業内で個別に開発するよりも、共同開発して共有する方が、テストの品質向上及び非競争領域への多重投資回避の面で有用である。特に、AUTOSAR OS におけるメモリ保護、タイミング保護、マルチコア拡張の検証は、知見が少なく、検証項目も膨大であるため、共同開発により 1 社あたりの開発投資を削減可能である。

我々は ATK2 のテストを、実施するべき内容から次の 5 種類に分類した。

- A) SS に着目したテスト
- B) OS 处理に着目したテスト
- C) マルチコア拡張に着目したテスト
- D) メモリ保護に着目したテスト
- E) タイミング保護に着目したテスト

本論文では、A)について述べる。

### 3. SS テストの概要

SS テストは、2009 年度から 2010 年度にかけて NCES で実施した API テスト[4]に対するテストスイート (TOPPERS Test Suite Package; 以降、TTSP と略す)をベースに行った。API テストのテスト対象は、シングルプロセッサ対応の TOPPERS/ASP カーネル(以降、ASP と略す)と、マルチプロセッサ対応の TOPPERS/FMP カーネル(以降、FMP と略す)である。ASP,FMP は ITRON 仕様[5]ベースの RTOS である。

SS テストとは、仕様に定められた振舞いが正しく実行されるかについて、SS 毎に確認するテストを指す。具体的には、SS 発行前のシステム状態(前状態)を定義し、その状態でテスト対象となる SS を発行(処理)し、SS 発行後のシステム状態(後状態)を確認する。これら一連のテストシナリオは後述する TESRY(TEst Scenario for Rtos by Yaml)記法によって記述される。TESRY 記法で記述されたテストシナリオデータ(以降、TESRY データと略す)は後述する AKTG によってテストプログラムに変換され、テストプログラムを実行することにより、テストを実施する。API テストの作業フローをベースにした SS テストの作業フローを図 1 に示す。

#### 3.1. テストシート

テストシートは、SS 每にテスト情報をまとめるものである。以下では TESRY データ生成のために特に重要なテストシートの構成情報であるテストケース、テストシナリオに関して述べる。

##### 3.1.1. テストケース

テストケースは SS 仕様を元に、SS の振舞いとして確認すべきテストケースを洗い出したものである。

図 2 に ActivateTask のテストケースの一部を示す。

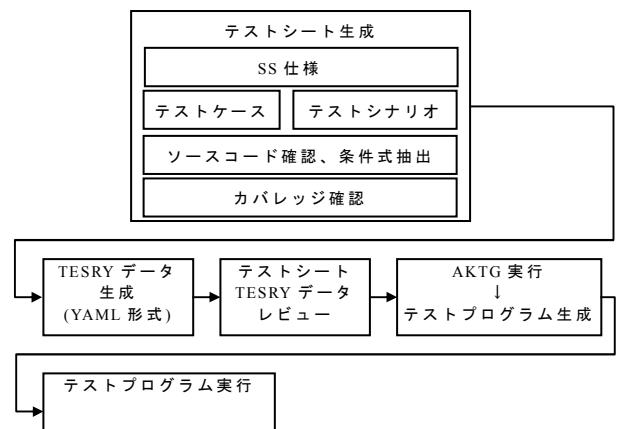


図 1. API テストの作業フローをベースにした SS テストの作業フロー

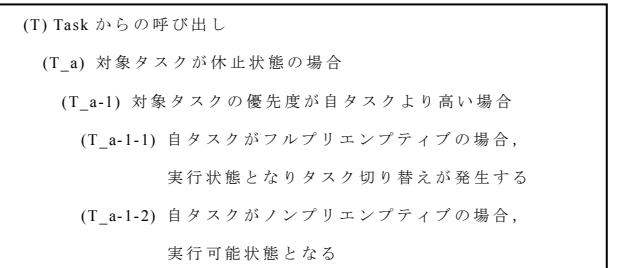


図 2. ActivateTask のテストケースの例

#### 3.1.2. テストシナリオ

テストシナリオとは、抽出したテストケース毎に、前状態、処理、後状態を日本語で記述し、TESRY データ生成のための情報をまとめたものである。図 2 のテストケース(T\_a-1-1)に対応するテストシナリオの例を図 3 に示す。

#### 3.2. TESRY データ

TESRY データはテストシナリオを階層的なデータ記述言語である YAML で表記したものである。

タスクを起動する SS である ActivateTask に対する TESRY データの例を図 4 に示す。TESRY データでは、最上位の階層として、前状態(pre\_condition), 処理(do), 後状態(post\_condition)が記述される。前状態と後状態の下位階層には、タスクなどのカーネルオブジェクトの状態が記述される。データの階層構造はインデントにより表現される。

前状態	
ID	: TASK1
タイプ	: タスク
状態	: 実行
優先度	: 中
スケジューリングポリシ:	フルプリエンプティブ
タイプ	: タスク
ID	: TASK2
優先度	: 高
状態	: 休止状態
処理	
TASK1 が ActivateTask(TASK2) を発行する。	
後状態	
タイプ	: タスク
ID	: TASK1
状態	: 実行可能
タイプ	: タスク
ID	: TASK2
状態	: 実行
優先度	: 高
起動回数	: 1

図 3. ActivateTask のテストシナリオの例

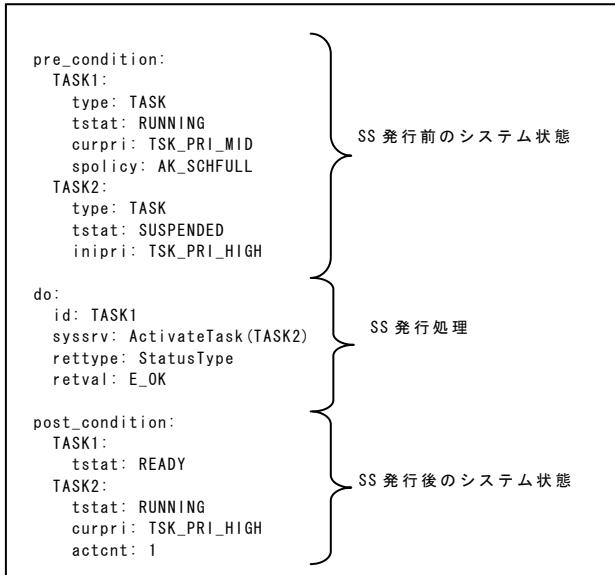


図 4. ActivateTask に対する TESRY データの例

### 3.3. AKTG

TTSP では、 TTG という独自に開発したテストツールを使って、 TESRY データからテストプログラムを生成していた。 TTG は ITRON 仕様ベースの RTOS 向けに開発されている。本研究で対象とする RTOS は AUTOSAR OS 仕様ベースであるため、 TTG を改変して AUTOSAR OS 仕様ベースに対応させた。対応させたツールは、 AKTG(Automotive Kernel Test Generator)と命名した。AKTG は前述の TESRY データを入力とし、 C 言語で記述されたテストプログラムを生成するツールである。AKTG はコメントを含めたプログラム行数が約 16,300 行の Ruby プログラムである。

## 4. テストスイート開発における課題

図 1 で示した作業フローで SS テストを実施しようとした場合、以下に述べる課題が発生すると想定された。

- (1)組み合わせパターンの増加に伴うテストケース数の増加
- (2)テストケース間の文言の不一致
- (3)テストシートと TESRY データの保守性
- (4)テストケースと TESRY データの一貫性
- (5)トレーサビリティ確保の必要性

課題(1)はテスト対象の RTOS のベース仕様が ITRON と AUTOSAR OS で異なることから発生する課題である。課題(2), (3), (4)は TTSP の既知の課題である。課題(5)は機能安全への対応から発生する課題である。以下に各課題の詳細を述べる。

### 4.1. 組み合わせパターンの増加に伴うテストケース数の増加

API テストでテスト対象とした ASP/FMP の API は、カーネル実行中に呼び出し可能なものが 104 個である。これに対し、SS テストでテスト対象とする ATK2 の SS は、カーネル実行中に呼び出し可能な 53 個である。しかし、ITRON ベースの ASP/FMP に比べ、ATK2 ではタスクなど各種オブジェクトで考慮すべきパラメータが増えており、それに起因する組み合わせパターンの急増が発生するため、テストケース数が増加することが予想された。

例えば、ASP でタスクとして考慮すべきパラメータは、タスク優先度、タスク状態の 2 つである。それらパラメータの取りうる値を全て組み合わせてテストケースを生成する場合でも、タスク優先度が 3 パターン、タスク状態が 6 パターンの合計 18 パターンである。対して、AUTOSAR OS 仕様ベースの ATK2 では、タスクとして考慮するパラメータは、タスク優先度 3 パターン、タスク状態 4 パターンに加え、タスクタイプ 2 パターン、スケジューリングポリシ 3 パターンが追加され、取りうる組み合わせの合計は 72 パターンまで増加する。さらに、ATK2 では、SC によってメモリ保護、タイミング保護の組み合わせパターンも追加して考慮する必要があるため、テストケース数がさらに増加すると予想された。

テストケース削減のために、HAYST 法などのテスト技法を適用すると、削減したテストケースで検出すべき欠陥が摘出できなくなるため、品質への影響が懸念された。よって、SS テストは全因子間での組み合わせテストを実施する方針とした。しかし、組み合わせを全網羅したテストケースを手動で生成することは困難

である。

#### 4.2. テストケース間の文言の不一致

API 每に担当者を割り振り、図 2 で示したように日本語を用いてテストシートの生成を行うので、同一内容のテストケースであっても、文言が一致しない場合があった。日本語の文言の不一致により、TESRY データが異なる内容になる危険性を排除するために、文言の統一が必要となった。

#### 4.3. テストシートと TESRY データの保守性

API テストでは、テストケースを識別するテスト ID は、各 API のテストケース毎に(a)からアルファベット順にテスト番号を付与するので、テストケースの増減が発生した場合に、テストケースや TESRY データのすべての番号を付与し直す必要があり、保守性が低下した。さらに、テストシートと TESRY データはテキストファイルで管理されているので、テストポリシーの変更があると、対応するすべての API への対応が必要となった。

#### 4.4. テストケースと TESRY データの一貫性

API テストではテストケースから実際にテストを実行するための TESRY データを生成するという作業を行っているので、テストケースに変更があると、TESRY データを修正する必要があり、修正忘れが起きるリスクが残された。結果的に、テストケースからテストプログラムまでの一貫性が取れない状況が発生した。

#### 4.5. トレーサビリティ確保の必要性

トレーサビリティとは仕様書間および、仕様書とソースコードなどの関連付けを実施するための仕組みである。外部仕様書ではトレーサビリティを確保するため、記載した仕様毎に仕様タグを割りつけている。仕様タグの種類には、NCES が仕様タグを割り当てた OSEK/VDX OS Version2.2.3 仕様と、AUTOSAR が仕様タグを割り当てた AUTOSAR OS Release4.0 仕様、さらに NCES が独自に追加し、仕様タグを割り当てた NCES 仕様が存在する。API テストの対象であった ASP/FMP の仕様を規定する統合仕様書には、仕様タグに相当するものが存在しなかったので、TTSP ではトレーサビリティの確保は実施していなかった。しかし、機能安全の観点からはトレーサビリティの確保は重要と考えられるため、外部仕様書には仕様タグを付与した。よって、テストケースとテスト対象の仕様のトレーサビリティ確保のために、生成する TESRY データに対しても、仕様タグを割りつける必要がある。

### 5. ツールに対する要件

上述の課題を解決するため、TESRY データの生成にツールの導入を検討した。以下に、ツールに必要な要件をまとめる。

- (1)組み合わせテストケースの生成
- (2)入力ファイルのモデル化
- (3)出力フォーマット
- (4)保守対象の局所化
- (5)仕様タグの付与

#### 5.1. 組み合わせテストケースの生成

課題(1)は、ツールによって、組み合わせテストケースを生成することで解決可能である。ただし、仕様によって組み合わせることができないパラメータが存在するため、ツールには組み合わせ不可能なパラメータが存在するテストケースを排除する機能が必要である。

#### 5.2. 入力ファイルのモデル化

課題(2)は、ツールの入力ファイルをモデル定義することで解決可能である。よって、ツールの入力ファイルは、API テストにおけるテストケース(図 2)のように非形式言語で記述されるものではなく、モデル化されていることが必要である。モデル化されたテストケースの構成要素をひとつのファイルにまとめて担当者間で共有することで、容易に意識合わせを実施可能である。

#### 5.3. 出力フォーマット

課題(3)は、ツールによって、入力ファイルから TESRY データの生成までを一貫して実行することで解決可能である。よって、ツールの最終的な出力フォーマット形式は、TESRY 記法に沿った YAML 形式(以降、TESRY 形式と略す)である必要がある。

これにより、入力情報の決定から、AKTG を使用したテストプログラムの生成までを自動化することが可能なので、入力ファイルからテストプログラム生成までを一貫して実行可能になる。

#### 5.4. 保守対象の局所化

課題(4)は、保守対象を TESRY データではなく、ツールの入力ファイルに局所化すると解決可能である。4.1節で述べたように、組み合わせテストケース数が膨大になった場合、保守対象となる TESRY データの数も膨大になると考えられる。よって、保守対象はツールの入力ファイルに局所化する必要がある。同様の理由で、TESRY データに対するテスト番号の付与はツールで実行する必要がある。

## 5.5. 仕様タグの付与

4.5節で述べたように、外部仕様書とテストケースの間でトレーサビリティを取るために、仕様タグを TESRY データに付与する必要がある。これは、5.3節により、ツールの出力データを TESRY データとした上で、さらに TESRY データへテストの対象とする仕様タグを付与することで解決可能である。図 5 に TESRY データにおける仕様タグ付与の例を示す。

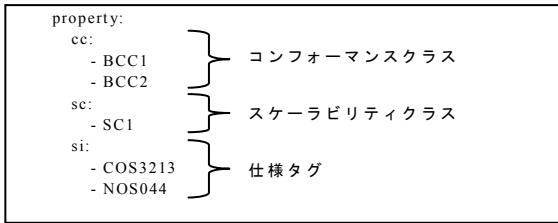


図 5. TESRY データにおける仕様タグ付与の例

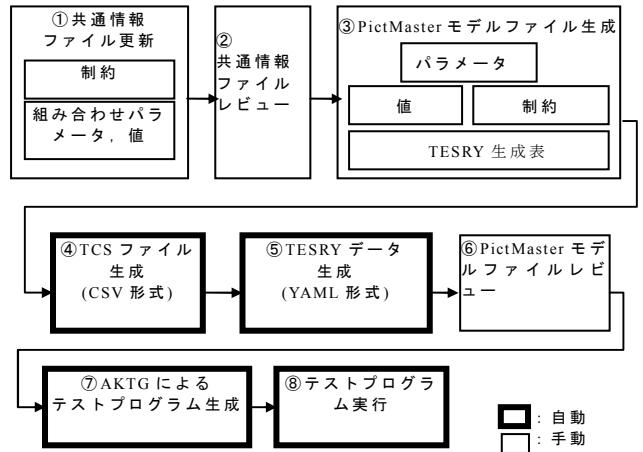
TESRY データにおいて、仕様タグはプロパティ層の下位層に定義される。コンフォーマンスクラス(以降、CC と略す)とは、OSEK/VDX OS 仕様で定義された OS 機能の部分的な実装を可能とすること等を目的とした、機能のサブセットの事である。

## 6. PictMaster

組み合わせテストツールとして、岩通ソフトシステム株式会社が開発したエクセルベースの組み合わせテストケース生成ツール PictMaster[8]が知られている。5 章で述べたツールに対する要件を、PictMaster が満たすか検討したところ、若干の拡張が必要であったが、要件を満たすことがわかった。さらに、PictMaster はオープンソースで公開されているため、拡張して利用することが容易である。SS テストでは、PictMaster を SS テストの用途に合わせ改変することで、前述の課題を解決した。

### 6.1. PictMaster を用いた SS テスト

図 6 は、PictMaster を使った SS テストの生成フローである。作業①～③では、PictMaster に入力する情報を作成する。作業④、⑤では、PictMaster 及び Ruby スクリプトを連携させることで、TESRY データを生成する。作業⑥では、AKTG の入力ファイルをレビューする。作業⑦では、AKTG によるテストプログラムの生成を行う。作業⑧では、テストプログラムを用いてテストを実行する。以下では、PictMaster による要件への対応と、作業の詳細を述べる。



### 6.1.1. PictMaster による組み合わせテストケース生成への対応

PictMaster は組み合わせテストケースの生成機能を備えている。これにより、PictMaster は組み合わせテストケースを自動生成する要件を満たす。更に、組み合わせ制約条件(以下、制約と略す)の規定により、特定の組み合わせパラメータを排除可能である。以上により、PictMaster は要件(1)を満たす。

なお、PictMaster には、Pairwise 法により組み合わせパターン数を削減する機能がある。しかし、今回の SS テストでは、4.1節で述べたとおり、全組み合わせのテストを実施する方針のため、Pairwise 法を用いたテストケース削減は実施しない。

### 6.1.2. PictMaster による入力ファイルモデル化への対応

PictMaster に入力する情報は、組み合わせパラメータ、値、後述する制約、TESRY 生成表である。これらは全てモデル化されている。よって、要件(2)は満たされる。

モデル定義に用いるパラメータ、値、制約は、異なる SS 間で、同じものを使用するケースがある。よって、SS テストの担当者間で、パラメータ、モデルのばらつきが生じるリスクがある。そこで、SS テスト共通情報ファイルを用意し、パラメータ、値、制約を異なる SS テスト間で共有可能にした(作業①)。

我々は PictMaster のエクセルファイルを SS 毎に用意し PictMaster モデルファイルと命名し、上述の入力情報をモデル定義した(作業③)。

### 6.1.3. PictMaster による TESRY データ生成および仕様タグ付与への対応

要件(3), 要件(4)を満たすために, PictMaster は入力情報から TESRY データ(TESRY 形式)を出力可能である必要がある. 更に, 要件(5)を満たすため, TESRY データのプロパティには, 仕様タグを付与可能である必要がある. 以下に, PictMaster を利用した TESRY データの生成に関して述べる.

TESRY データは, 前状態, 処理の内容が決定すると, プロパティ, 後状態は前状態, 処理の内容と, 外部仕様から一意に決定される特性を持つ. つまり, TESRY データの前状態, 処理の部分で組み合わせテストの対象とするパラメータ, 値を定義すれば, 前状態で使用する TESRY データの構成パラメータ(以降, TESRY パラメータと略す)は組み合わせパターンの構成値から決定可能である. 加えて, 外部仕様から後状態およびプロパティ(仕様タグを含む)の TESRY パラメータも決定可能である.

SS テストでは, 組み合わせパターンの値から, TESRY パラメータを決定, 出力するために, PictMaster の期待結果出力機能である結果表を拡張し, TESRY 生成表と命名し, 利用した. 図 7 に後状態の TESRY パラメータの決定イメージを示す.

以上の対応により, PictMaster に組み合わせパラメータ情報と, TESRY 生成表を入力し, 組み合わせパターンの一覧と, TESRY パラメータが記載されたファイルを出力させた(作業④). 我々は出力ファイルを TCS(TestCaseSheet)ファイルと命名した. TCS ファイルのフォーマット形式は CSV 形式である. TCS ファイルでは, 1 つの行が 1 つのテストケースに相当する. 図 8 に ActivateTask の TCS ファイルから抽出した 1 行を示す.

各行の最初のフィールドは, テストケース番号の生成に用いる連番を示す, この番号は PictMaster の機能で自動的に付与される. 続くフィールドは組み合わせパターンを構成する値の羅列である. 図で太字かつ二重引用符で囲まれたフィールドが, TESRY パラメータを羅列した部分である.

TCS ファイルは CSV 形式であり, 最終的な期待出力である TESRY 形式に変換する必要がある. そこで, 160 行程度の Ruby スクリプトを作成し, TCS ファイルから TESRY データを自動生成した(作業⑤).

以上で述べたように, PictMaster を拡張してテストケースに相当する入力情報から TESRY データを自動生成可能になったことで, 要件(3)は満たされる. また, 出力ファイルのフォーマット形式は TESRY 形式であることから, 要件(4)は満たされる. また, 仕様番号と対応する組み合わせパターンの条件を TESRY 生成表

に入力すれば, 出力される TESRY データに対して仕様タグを付与可能になった. よって要件(5)は満たされる.

組み合わせパターン	
バラメータ (意味)	値 (意味)
other_inipri (SS 発行対象タスクの初期優先度)	TSK_PRI_HIGH (自タスクより高い)
other_tstat (SS 発行対象タスクの状態)	SUSPENDED (休止状態)
origin_spolicy (SS 発行元タスクのスケジューリングポリシー)	AK_SCHFULL (フルブリエンブティブ)

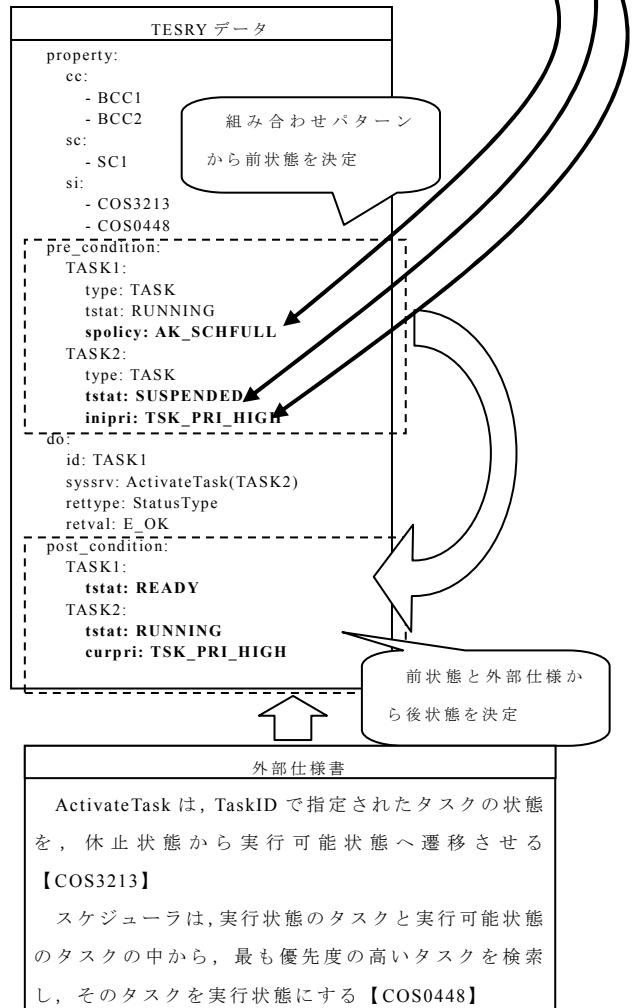


図 7.後状態の TESRY パラメータの決定イメージ

```
7,C2ISR,READY,TSK_PRI_MID,AK_BASIC,NON_EV
T,AK_SCHFULL,1,ORG_TASK,ENA_ALL_INT,"Activat
eTask_normal_0007,property,cc,BCC2,ECC2,sc,si,NOS
044,COS3104, ,pre_condition,TASK1,type,TASK,tstat,R
UNNING, ,(以下のフィールドは省略)"
```

図 8.TCS ファイルのサンプル

## 7. 実施結果および評価

### 7.1. 実施結果

2011 年度上期終了時点で、PictMaster を用いて約 3,500 件の TESRY データを自動生成した。その後、AKTG を用いてテストプログラムを自動生成し、テストを実施した。テスト実施により検出した ATK2(SC1) の実装における不具合は 7 件であった。具体的には、SS の返り値が仕様と異なる、不正な処理単位から SS を呼出した場合にエラー検知しない、などの不具合を検出した。

### 7.2. 課題の解決

PictMaster の組み合わせテストケース生成機能を利用し、組み合わせパターンの増加によるテストケース数の増加によるテストケース生産コストの増加を低減した。

PictMaster モデルファイルにテストケースに相当する内容をモデル定義した。これにより、日本語によるテストケースのばらつきを排除した。

PictMaster の VBA に実プログラム行数で 50 行程度の拡張を加え、TESRY データの自動生成を実現した。これにより、テストケースに該当する PictMaster モデルファイルと TESRY データの一貫性を確保した。

保守対象を共通情報ファイル及び PictMaster モデルファイルに集約した。これにより、TESRY データを直接保守する必要性を排除し、AUTOSAR OS 仕様変更時の保守性を向上した。

TESRY データに対して自動的に仕様タグを振り分けた。これにより、TESRY データと外部仕様書のトレーサビリティを確保した。

### 7.3. 評価

作成した TESRY データの総行数は約 185,400 行になり、TTSP のように手動で生成、保守するのは困難である。よって、TESRY データの生成を PictMaster の導入と 50 行程度のプログラム拡張、160 行程度の Ruby スクリプトの作成によって自動化できた事は、大幅な作業コストの削減に繋がった。また、保守対象を TESRY データから PictMaster モデルファイル、共通情報ファイルに局所化し、レビュー、保守コストを削減することができた。共通情報ファイルに記載する制約は、共通のパラメータ、値を用いる SS 間で共有できたので、PictMaster モデルファイルの生産性向上に繋がった。

テストで検出した不具合には、SS の呼出し元など、多くの値を取るパラメータの組み合わせを PictMaster で全網羅した結果、発見できた不具合も含まれた。

## 8. 今後の展開、課題

### 8.1. TOPPERS ASP/FMP カーネルへの適用

TOPPERS ASP/FMP カーネルで実施した API テストにフィードバックし、網羅性の確保によるテスト品質の向上を検討する。

### 8.2. SS テスト以外のテストへの適用

同様の組み合わせテスト技法が SS 以外のテストに対して、適用可能性について検討する。

### 8.3. 外部仕様書と PictMaster 間のトレーサビリティ手法を検討する

SS テストでは、外部仕様書に振られた仕様タグと、PictMaster モデルファイルに記載する仕様タグの関連付けは手動で実施した。これにより、仕様タグが大幅に変更された場合、人間が目で変更分の影響範囲を追うことになり、生産性が下がった。この問題に対応するため、外部仕様書とモデルファイル間でトレーサビリティを自動で確保するためのツールの作成等を検討する必要がある。

**謝辞** 本手法の検討に参加し、さらに論文執筆にご協力下さった NCES 研究員の皆様に謹んで感謝の意を表する。NCES ではコンソーシアム型共同研究に参加を希望する企業・団体を募集している。

## 文 献

- [1] AUTOSAR,  
<http://www.autosar.org/>
- [2] OSEK VDX,  
<http://www.osek-vdx.org/>
- [3] 名古屋大学大学院情報科学研究科附属組込みシステム研究センター,  
<http://www.nces.is.nagoya-u.ac.jp/>.
- [4] 鳴原一人, 松浦光洋, 金ハンソル, 金スンヨプ, 馬銳, 廉正烈, 金榮柱, 木村貴寿, 真弓友宏, 本田晋也, 山本雅基, 高田広章, 組込みリアルタイム OS に対する API テストの実施, ソフトウェアテストシンポジウム 2010 予稿集, (pp.46--53, 2010)
- [5] 坂村健監修, 高田広章編, “ $\mu$  ITRON4.0 仕様 Ver.4.02.00”トロン協会, 2004.
- [6] 鳴原一人, 森孝夫, 本田晋也, 山本雅基, 高田広章, RTOS のテスト自動生成システムに関する一考察, 情報処理学会 組込みシステム(EMB), (Vol.2010-EMB-16 No.11, pp.1--8, 2010)
- [7] 鳴原一人, 真弓友宏, 本田晋也, 高田広章, マルチプロセッサ対応 RTOS を対象としたテストシナリオ記述法とテストプログラム生成ツール, 情報処理学会 組込みシステム(EMB), (vol.2010-EMB-18 No.1, pp.1--8, 2010)
- [8] PictMaster,  
<http://sourceforge.jp/projects/pictmaster/>