

相互運用性試験に対する試験系列生成方式の検討

坪根 宣宏 土岐田 義明
(株)高度通信システム研究所

筆者らは相互運用性試験のための試験系列生成を機械的に支援するための方式を検討し、提案してきた。従来提案した方式では、試験を実施する2つの試験対象のプロトコル仕様をそれぞれ有限状態機械(FSM)でモデル化し、さらにその2つのFSMの各状態を合成して得られるシステム状態グラフ(SSG)を導出した後、そのSSGをトレースすることにより試験系列を生成する。本方式の問題点はFSMの定義の仕方により容易にSSGが状態爆発を起こして有限の試験系列が得られずこれが実用上の障害であった点である。本論文では、このような状態爆発を防ぐ手段、及び筆者らが開発した試験系列生成プログラムの概要について述べる。

Study of a Test Suite Generation Method for Interoperability Testing

Nobuhiro TSUBONE and Yoshiaki TOKITA

Advanced Intelligent Communication System Laboratories

We have proposed some of test suite generation methods for interoperability testing. In our conventional method, first of all, each IUT(Implementation Under Test) is modeled as an FSM(Finite State Machine). Then an SSG(System State Graph) that is acquired by synthesizing each state of the two FSMs is derived and a test suite is generated by tracing the SSG. The problem we think is the SSG may easily cause the state explosion, which is an obstacle in practical testing. This paper states how to avoid the state explosion and an overview of the test suite generation program we have developed.

1. はじめに

昨今の情報通信システムのソフトウェアは、提供するサービスの多様化やオープン化指向に伴い、大規模化、複雑化の一途を辿っている。システムの品質を確保するためには、そのソフトウェアに対して信憑性の高い試験の実施が不可欠である。試験の信憑性は、網羅性の高い試験系列を如何に生成する

かに関わっている。更には、開発コストの観点から言えば、その効率も重要である。

従来、試験系列生成の研究は適合性試験に対するものが多く、相互運用性試験に対するものは少なかった。その中で、当研究所では、相互運用性試験に関わる2つのIUT(Implementation Under Test; 試験対象)をFSM(Finite State Machine; 有限状態機械)でモ

デル化し、その2つのFSMの各状態を合成して得られるSSG(System State Graph; システム状態グラフ)の導出を経由して試験系列を生成する方式[1]を提案した。本方式の問題点は、FSMの定義の仕方により、それらのFSMから導出されるSSGが容易に状態爆発を起こして、有限な試験系列が生成できず、結果として実施可能な試験系列が得られないことである。

本論文では、まず従来方式の概略を述べた後、上記の問題点を検討し、1つの解決策について示す。更に、筆者らは本解決策を反映した試験系列生成プログラムを開発したのでここに紹介する。本プログラムはFSMの形式で各IUTのプロトコル仕様を入力すると、国際標準の試験記法であるTTCN(Tree and Tabular Combined Notation) [2] のMP(Machine Processable)形式で記述された試験系列を出力する。

2. 従来の試験系列生成方式

2.1 試験環境のモデル

従来の試験系列生成方式では、図2.1に示す試験環境モデルを想定している。各IUTはFSMの形式でモデル化する。さらに、各IUTには、試験イベント(単一の試験データ)を制御・観測する拠点であるPCO(Point of Control and Observation)を想定し、IUTの上部にあるものを「上位PCO」、下部にあるものを「下位PCO」と呼称する。試験イベントは各PCO対応に存在するテストにより制御・観測される。channel12、channel21はOSI参照モデルにおける下位層のサービスプロバイダをモデル化したものであり、ここではFIFO(First In First Out)キューを想定している。試験系列を構成する各試験ケースはこれらの試験イベントの羅列(正確には、分岐を考慮した木表

現)であり、各テストが対応するIUTとの入出力を通して試験を実施するための詳細な手順を示している。本方式では、このようなテストの存在を仮定し、そのテストのための試験系列を出力することを目的とする。

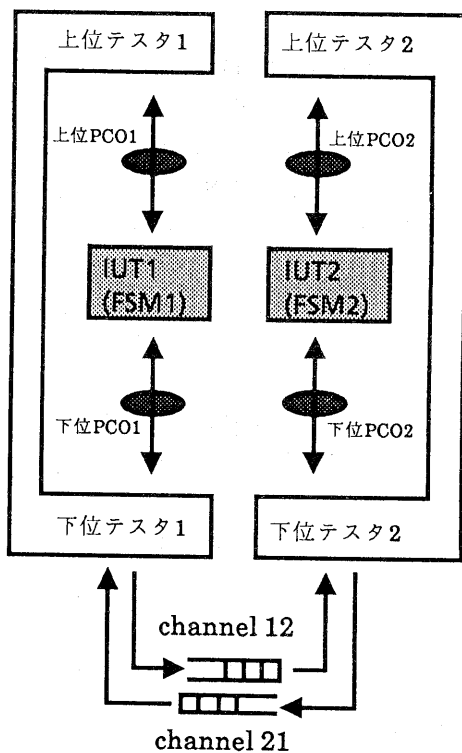


図2.1 相互運用性試験に対する環境モデル

2.2 試験系列の生成方式

試験系列は、図2.1の環境モデルにおける2つのIUTをモデル化した各FSMを合成して得られるシステム状態を導出し、それらの状態の集合をグラフとして表現したSSGを初期状態からトレースすることにより試験系列を生成する。SSGは次のように定義できる。

[SSGの定義]

FSM1、FSM2、channel12、channel21における各状態を組み合わせてできる全状態のうち、初期状態から到達可能な状態の集合で構成されるグラフ。

ここで、SSG合成の一例を図2.2に示す。本図で、(A)、(B)はある2つのIUT、IUT1とIUT2の仕様をFSM形式でモデル化した状態遷

移図形式で表現したもの、(C)は(A)、(B)の仕様を合成して導出されるSSGである。図中、U1?CONreq/L1!CR、L2?CR/U2!CONindなどの表記はIUTに対して入出力されるイベントを表す。U1?CONreq/L1!CRは、IUT1の上位PCO(U1)からイベントCONreqがIUT1に入力されるとIUT1の下位PCO(L1)へイベントCRを出力することを表し、L2?CR/U2!CONindはIUT2の下位PCO(L2)からイベントCRが

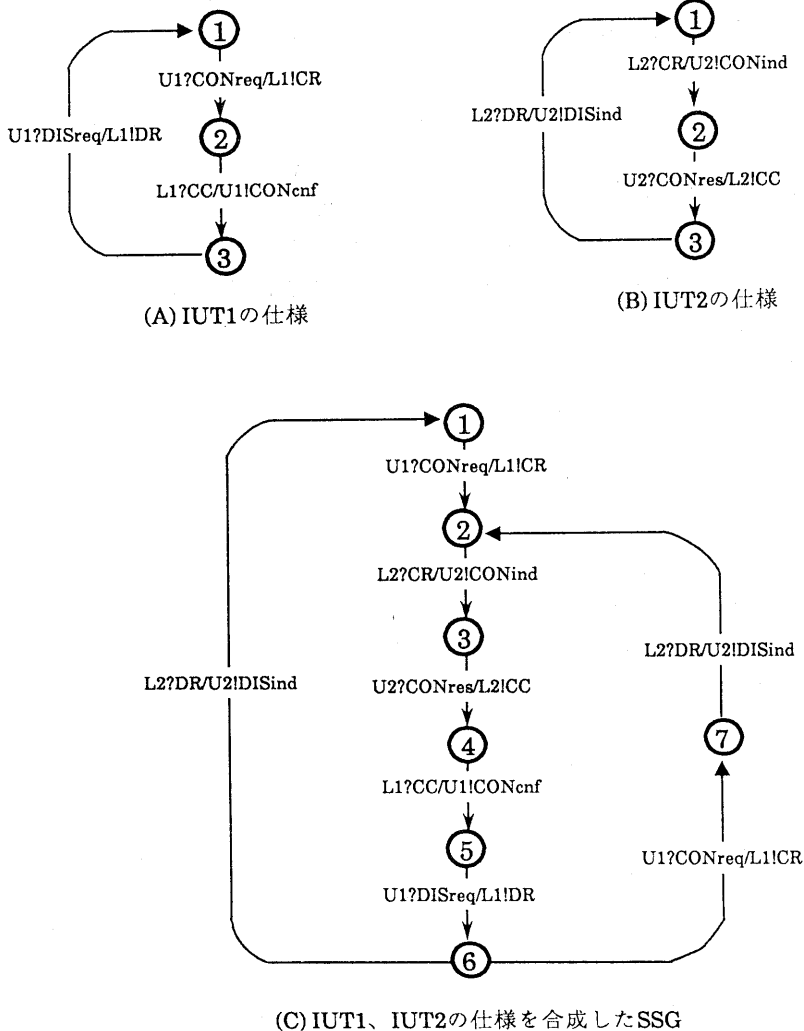


図2.2 SSG合成の一例

IUT2に入力されるとIUT2の上位PCO(U2)へイベントCONindを出力することを表す。他のイベントも同様の意味を持つ。

相互運用性試験の試験系列を生成するより簡易な方法としては、FSM1、FSM2を互いに独立にトレースする方法もありうる。しかし、この場合にはFSM1に属するイベントとFSM2に属するイベントの順序関係が不明になるという問題点がある。一方、本方法では、FSM1、FSM2を合成して得られるシステム状態グラフの導出を經由して試験系列を生成するためFSM1、FSM2相互のイベントの順序関係もその試験系列の中で明示的に示すことができる。

3. 従来方式の問題点と解決策

従来の方式では、図2.1に示す試験環境のモデルにおいて下位層のサービスプロバイダをモデル化したchannel12、channel21のキューに何らの制限を設けていなかった。両キューの状態はキュー長とキューイングされるイベントの種類によって識別されるため、キュー長に制限がない場合、キューの状態は一般的には無限に存在する。従って、FSM1、FSM2、channel12、channel21各状態の組合せで構成されるシステム状態の数も無限に存在することになり、SSGが状態爆発を起こす可能性がある。例えば、FSM1において同一状態で同一イベントを繰り返し送信し、FSM2においてそのイベントを同一状態で繰り返し受信するような遷移が存在すれば、channel12には無限にそのイベントが滞留していくため、対応するSSGが状態爆発を起こすのは明らかである。

筆者らは、上記のような状態爆発を防ぐため、channel12、channel21のキュー長に制限を設けることにした。すなわち、SSGを導出

するためのアルゴリズムにおいて、そのキュー長を越えてイベントがキューイングされる事態が検出された場合には、そのイベントを廃棄する手順を追加するようにした。

キュー長に制限を設けることは、生成される試験系列に対する一種の試験範囲の選択手段であると考えられる。実際のキュー長の値は、IUTの仕様や試験の規模などに応じて適宜調整する必要がある。

4. 開発した試験系列生成プログラム

筆者らは前述した方式を実装した試験系列生成プログラムを開発した。本プログラムはSun SPARCstationのOpenWindows上で動作し、下記の機能を持つ：

- 図2.1に示すモデルに基づく相互運用性試験用試験系列生成機能
- ISO9646-1[3]に示されるモデルに基づく適合性試験用試験系列生成機能
- TTCN MP形式に基づく試験系列記述機能
- FSM、SSGの状態遷移図形式による表示機能
- チャネル容量(キュー長)の設定機能
- SSG生成時のシステム状態数の最大数(状態爆発の検出用)設定機能

本プログラムによる操作画面例を図4.1に示す。図4.1中の右側にある“TESGEN”ウィンドウは本プログラムのメインウィンドウであり、その下部サブウィンドウに本例で生成したSSGを状態遷移図形式で表示している。また、その上部サブウィンドウには、上記SSG表示で反転表示された状態“2”の詳細情報を表示している。図4.1で左側にある“Generate”のサブウィンドウは本操作の入力であるFSM形式によるIUTの仕様ファイル(IUT1、IUT2)、

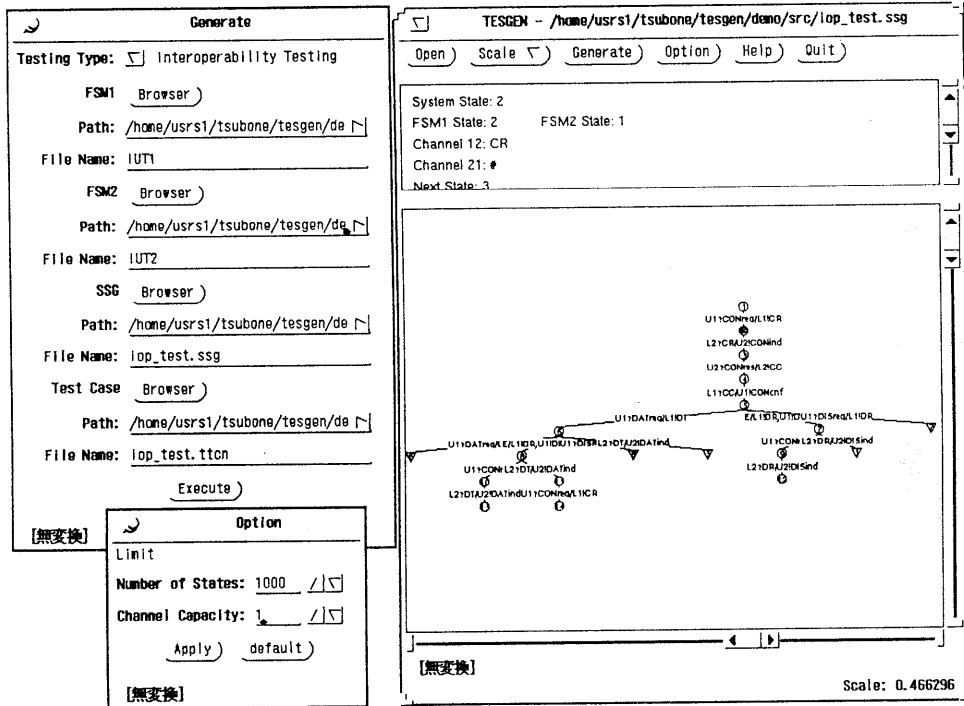


図4.1 筆者らが開発した試験系列生成プログラムの操作画面例

及び出力であるSSGファイル(iot_test.ssg)、試験系列ファイル(iot_test.ttcn)を指定するために用いる。“Option”のサブウィンドウは、Number of States(システム状態数の最大数)とChannel Capacity(チャンネル容量)を設定するために用いる。なお、本操作時の入力であるFSM形式によるIUTの仕様は、一般のテキストエディタによるテキストファイルとして作成する。本例における各IUTの仕様(FSM形式)は次の通りである。

[IUT1の仕様]

- 1, 2, U?CONreq, L!CR.
- 2, 3, L?CC, U!CONcnf.
- 3, 3, U?DATreq, L!DT.
- 3, 1, E, L!DR, U!DISInd.
- 3, 1, U?DISreq, L!DR..

[IUT2の仕様]

- 1, 2, L?CR, U!CONInd.
- 2, 3, U?CONres, L!CC.
- 3, 3, L?DT, U!DATInd.
- 3, 1, L?DR, U!DISInd..

上記の仕様で、例えば“1, 2, U?CONreq, L!CR.”は状態1において上位PCOからIUT1にイベントCONreqが入力されると下位PCOにCRを出力して状態2に遷移することを意味する。本プログラムで生成する試験系列はTTCN MP形式におけるDynamicPartの構文に基づくテキストファイルとして出力する。なお、試験系列で扱うイベントはテストの立場から見るため、その入出力の表記はIUT仕様の場合とは反対の表記となる。例えば、IUT1の仕様におけるU?CONreqは、試験系列の中

ではU1!CONreqと表記される。図4.1の操作で生成した試験系列ファイル(iop_test.ttcn)の内容は次の通りである。

[生成した試験系列ファイルの内容]

```
$DynamicPart
$TestCases
  $Begin__TestCase
    $TestCaseId
    $TestGroupRef
    $TestPurpose
    $DefaultsRef
    $Comment
    $BehaviourDescription
    $BehaviourLine
    $LabelId
    $Line [0] U1!CONreq
    $Cref
    $VerdictId
    $End__BehaviourLine
    $BehaviourLine
    $LabelId
    $Line [1] L1?CR
    $Cref
    $VerdictId
    $End__BehaviourLine
    $BehaviourLine
    $LabelId
    $Line [2] L2!CR
    $Cref
    $VerdictId
    $End__BehaviourLine
    .
    .
    .
$End__DynamicPart
```

5. おわりに

以上本論文では、従来の相互運用性試験のための試験系列生成方式に対して、状態爆発を防ぐ手段、及びその手段を実装した試験系列生成プログラムの概要について述べた。

今後の課題としては次の3つが挙げられる。

- (1) 本論文の方式で生成したATS(Abstract Test Suite) をETS(Executable Test Suite)に変換する際の機械的支援策の検討
- (2) 実際の試験実行ツールとの連携性評価
- (3) 本論文の方式で生成した試験系列に対して、実施可能な試験範囲を効率的に選択する手法の確立

[謝辞]

本研究に当たりご指導をいただいた(株)高度通信システム研究所 顧問 野口 正一氏(東北大学 名誉教授)に深謝いたします。また、本研究の機会を与えていただいた同研究所 緒方 秀夫常務に謝意を表します。さらに、本論文の執筆に当たり、貴重な意見を寄せてくれた(株)高度通信システム研究所の諸氏に感謝します。

[参考文献]

- [1] 朴 美娘、岡崎 直宣、太田 正孝、高橋 薫、白鳥 則郎、野口 正一:「プロセスの独立性を考慮した通信システムにおける相互接続試験系列生成法」、信学技報、IN92-20、pp.7-12 (1992)
- [2] ISO : “ISO 9646-3: Information processing systems --- Open Systems Interconnection --- OSI Conformance Testing Methodology and Framework --- Part 3: The Tree and Tabular Combined Notation (TTCN), ” (1991)
- [3] ISO : “ISO 9646-1: Information processing systems --- Open Systems Interconnection --- OSI Conformance Testing Methodology and Framework --- Part 1: General concepts,” (1991)

(注) 本論文中のSun、SPARCstation、OpenWindowsの表記はいずれも米国Sun Microsystems, Inc.の登録商標です。