2．通信プログラムの試験

Testing for Communication Programs by Fumiaki SATO（Faculty of Information, Shizuoka University）.

佐藤　文明[1]

1 静岡大学情報学部

1．はじめに

通信プログラムとは，その上位に存在する実際の応用プログラムに対して，下位の複雑な通信機構を隠蔽して，簡潔なインタフェースによって適切な通信機能を提供するためのソフトウェアである．通信プログラムは，複数の計算機に搭載され，通信プロトコルと呼ばれる手順に従ってメッセージの送受信や誤り訂正，メッセージの加工などを行っている．その意味で，通信プログラムは，通信プロトコルの仕様を実現したものと考えることができる．

通信プロトコルは，通信する計算機などのメッセージの通信手順を規定したものだが，国際的な標準化が行われている．とくに，OSI（開放型システム間相互接続）と呼ばれるISOの標準では，通信プロトコルを7階層にモデル化し，各階層におけるプロトコルおよび，上位に提供するサービスの規定が行われている．

また，ISOでは，OSIの通信プロトコルを規定するとともに，OSIの規定に整合しているかどうかを試験する方法をも規定している．これを適合性試験と呼ぶ[1]．適合性試験の標準には，試験の実施方法，試験環境，試験仕様の構成などについても言及されている．以下の章で，その概要について述べる．適合性試験をパスしても，実際に通信システム同士が相互接続できる保証はない．そのため，通常は適合性試験の後に，相互運用性試験を行う．以下の章では，相互運用性試験の目的，その構成について述べる．

さらに，ISOでは，通信システムの仕様を形式的に記述するための技法を標準化した．LOTOS[2]やEstelle[3]はISOによって標準化された形式記述言語である．形式的な手法によって記述された仕様からは，計算機処理によって仕様自体の検証，試験仕様の導出，プログラムの導出などが可能となる．また，適合性試験の枠組みを形式的に定義し，試験のカバー率の定義などを行っている．以下の章で，適合性試験の形式的な定義と，形式仕様からの試験データを抽出方法について述べる．

以下，第**2**章は通信プログラムの試験の特徴について述べる．第**3**章では，通信プログラムの試験構成について，とくに適合性試験，相互運用性試験，そして形式的なアプローチについて述べる．第**4**章では，通信プログラムの試験に必要な試験系列の生成方法について述べる．第**5**章は，まとめと今後の動向について述べる．

2．通信プログラムの試験の特徴

通信プログラムの試験は，通常の並列プログラムと異なり，次のような特徴をもっている．

(1) ブラックボックスで試験することが多い．

つまり，プログラムを作成する組織と試験を実施する組織とが異なり，プログラムの内部に触れないことがある．したがって，仕様から得られる情報のみで試験を行うことが多い．そのため，イベントのパラメタとその出現順序のみが試験仕様であり，試験結果である．

(2) 外部から制御できるインタフェースが限定される．

実際の動作環境で行われる試験では，詳細な点まで外部から確実に制御できないことがある．したがって，試験の対象が制限される場合もある．

(3) プログラムの仕様を形式的に与えやすい．

通信プログラムの動作は，メッセージの送受信である．その仕様は，基本的にはタスクの状態とタスク間のメッセージ交換の順序から規定される．仕様を形式的に記述する方式には，状態遷移モデルや，イベント順序論理モデルなどを使った多くの研究が行われており，また仕様記述言語もISOによって標準化されている．

このような通信プログラムの試験を行うためには，いくつかの段階を経て試験が行われることが多い．まず適合性試験と呼ばれる通信プログラムの仕様との整合性をチェックする試験を行う．次に，相互運用性試験と呼ばれる，通信プログラム間を接続して相互運用

が正しく行えるかどうかを確認する試験が行われる．最後に，性能試験や耐久性試験を実施して実運用が行われる．

3. 試験の構成

3.1 適合性試験[1)]

適合性試験の目的は，通信プログラムの相互運用性を高めるために行われる試験であり，通信プログラムの仕様から抽出された試験仕様に基づいて行われるものである．適合性試験では，通信プログラムが正しく動作することを保証するものではなく，通信プログラムに含まれる未発見の誤りを多く発見することが目的である．

3.1.1 適合性試験の手順

適合性試験では，まず静的な適合性について検査を行う．これは，通信プログラムが仕様のどのようなオプションを実装しているか，どのようなパラメタを選んで実装しているかを確認するもので，書面で行う．

次に，動的な適合性について検査を行う．これは，実際にテスタと試験対象（IUT: Implementation Under Test）とを接続して行う試験で，試験データは申告された仕様のパラメタに応じて選択されたものを使う．また，試験対象の性質（可搬性や可制御性）に応じた，適切なテスタと試験対象との構成を使って行われる．

この試験の結果は報告書として，試験者（試験センタ）から受験者に通知される．この結果に基づいて，認定機関が仕様への適合性を証明する認定書を発行するなどが行われる．

3.1.2 適合性試験の構成

適合性試験の概念的な構成では，テスタが試験対象へのアクセスポイントSAP（Service Access Point）を通して，ASP（Abstract Service Primitive: 抽象サービスプリミティブ）によって制御観測を行う．テスタと試験対象とは，プロトコル仕様に規定されたデータ構造をもつPDU（Protocol Data Unit: プロトコルデータ単位）を交換する．試験対象とテスタの配置によって，次の4つの構成が定義されている（**図-1**）．

(a) ローカル試験法

ローカル試験法は，テスタが直接試験対象の上位のインタフェースと下位のインタフェースを制御するもので，最も制御可能範囲が広い．テスタの内部には，論理的にIUTの上位と下位のインタフェースを制御観測するための上位テスタ（UT: Upper Tester），下位テスタ（LT: Lower Tester）が含まれている．

(b) 分散試験法

分散試験法は，遠隔のIUTの上位インタフェースを直接制御できず，IUTの上位に上位テスタのモジュールが設置されたり，オペレータが直接IUTを操作する必要がある．上位テスタと下位テスタとの間には，試験調和手順が規定されるが，それは通信プロトコルに限らずオペレータ同士の電話連絡であってもよい．

(c) 調和試験法

調和試験法では，UTが設置されIUTを完全に制御観測する．UTとLTとの間には試験調和手順が規定され，その実現として試験管理プロトコルという試験用の通信プロトコルが用意される．試験管理プロトコルによって，UTを制御したり，UTが受信したデータをLT側に転送する．

(d) 遠隔試験法

遠隔試験法では，IUTの上位は制御観測を行わない．したがって，下位インタフェースを通しての制御観測が可能な試験範囲に限定される．

応用に近い通信プロトコルでは，複数の相手との通信を規定しているものが存在する．たとえば，分散トランザクション処理を実現するためのOSI-TPプロトコルでは，複数のトランザクション処理要求を異なる通信相手に送信して，その応答によって処理が変わる複雑なプロトコルとなっている．このようなプロトコルを試験するための構成として，マルチパーティ試験が規定されている．**図-2**は，マルチパーティ試験の構成である．テスタは，複数のコネクションに対応する個別のテスタと，それを制御する管理用テスタによって構成される．

3.2 相互運用性試験[4)]

相互運用性試験は，適合性試験などで十分なプログラムの品質の高さが確認された通信プログラムを，相互に接続して動作確認を行うためのテストである．相

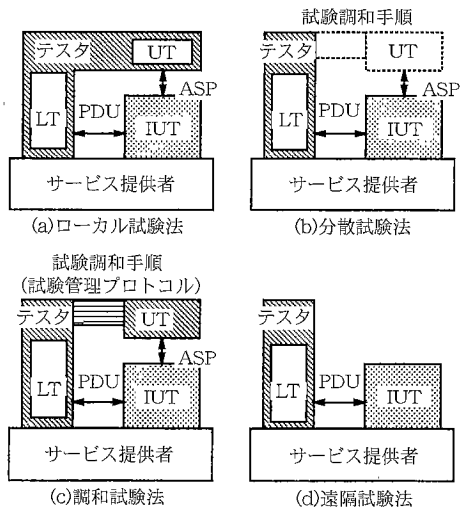


図-1 適合性試験の構成

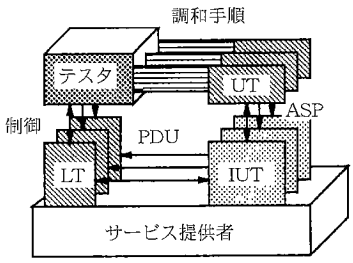


図-2　マルチパーティ試験の構成

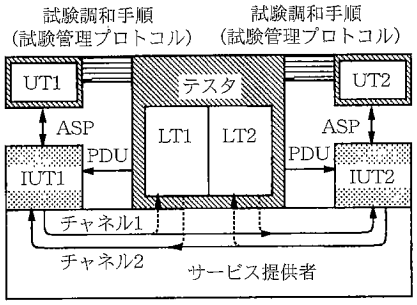


図-3　相互運用性試験の構成

互運用性試験では，複数の通信プログラムが実際に通信を行う試験が行われるが，それによって適合性試験では確認されなかった実際のタイミングでの動作試験が行われる．

相互運用性試験は，標準化は行われていない．相互運用性試験の構成としては，文献4）によって想定されている構成を**図-3**に示す．ここで，IUT1とIUT2とは下位テスタLT1とLT2を通して通信を行っている．IUT1とIUT2は有限状態機械でモデル化されており，これらの2つのIUTの仕様から，相互運用動作を検査する試験データを導出し，それを使って試験を行う．LT1とLT2は，サービス提供者上に確立されているIUT間の通信チャネルのタイミングを制御することも行う．

文献4）では，はじめにこの2つの有限状態機械，および2つの通信チャネルの状態を合成してシステム状態グラフを作成する．そして，システム状態グラフの各状態をたどる制御手順を求めて，その手順に従うようにテスタを動作させ試験を実施する．

このシステム状態グラフは，通常膨大になるため，すべての状態遷移を検証することができない．そのため，適切な試験範囲を選択する必要があるが，人手による選択は状態数の大きさから困難である．そのための自動的な相互運用性試験データの選択アルゴリズムが開発されている．

3.3　形式的アプローチ

形式的なアプローチの目的には，試験の自動化やプログラムの品質を客観的に評価することにある．通信プログラムの仕様を形式的に記述することから，試験データの自動生成や適切なコストと品質の試験が可能になる．また，試験の枠組みを形式的に定義することで，適切な試験系列生成のアルゴリズムの選択や，試験環境の選択が可能となる．

ISOでは，通信システムの仕様を記述するための言語として，LOTOS[2]とEstelle[3]という言語を標準化してきた．それぞれ各言語は，基盤としてCCSやCSPといったプロセス代数や抽象データ型，拡張有限状態機械などの数学的なモデルが使われている．これらの言語から，さまざまな目的の試験データを導出するためのアルゴリズムやシステムが研究されてきている．

また，同時にISOでは適合性試験の形式的な定義をも行っている[5]．現在，そのプロジェクトは，国際標準の一歩手前の段階であり，あと1年程度で国際規格として標準化が完了する．

4. 試験系列の自動生成

試験系列は，テスタによって試験対象に入力されるべきイベント，試験対象から得られるべき出力イベントの系列のことである．形式的な仕様からの試験系列の自動生成には，さまざま方法があるが，ここでは有限状態機械（Finite State Machine: FSM）に基づく方法と，イベント順序論理（Temporal Ordering Logic）に基づく方法について述べる．

4.1　有限状態機械に基づく方法

はじめに，ここで扱う有限状態機械は完全定義であり，冗長な状態が存在せず，強連結であると仮定する．また，仕様自体には誤りがないと仮定する．

有限状態機械は，$<S, I, O, \delta, \gamma, s_i>$のように定義される．ここで，Sは状態集合，Iは入力集合，Oは出力集合，δは出力関数，γは遷移関数，s_iは初期状態である．たとえば，仕様Mの遷移表が**表-1**のように与えられた場合，状態集合，入力集合，出力集合は次のように表される．

$S = \{s1, s2, s3, s4, s5\}$
$I = \{i1, i2\}$
$O = \{o1, o2\}$

このとき，各状態に定義された出力（出力関数）および状態遷移（遷移関数）を検査することが，試験系列の目的である．

4.1.1　トランジションツアー法

Naitoらは，トランジションツアー法と呼ぶ有限状態機械からの試験系列生成法を提案した[6]．この方法は，状態遷移表のすべての欄を通過する系列を求める手法であり，現在広く行われている試験も基本的には

表-1　Mの状態遷移表

state	s1	s2	s3	s4	s5
i1	o2/s5	o2/s1	o2/s2	o1/s2	o1/s3
i2	o1/s4	o1/s4	o2/s2	o2/s1	o2/s3

表-2　Mの実現（M1）

state	s1	s2	s3	s4	s5
i1	o2/s5	o2/s1	o2/s2	o1/s2	o1/s3
i2	o1/s4	o1/s4	o2/s2	o2/s1	o2/s1

表-3　DS（i1i1i1i1）を与えたときのMの出力

state	output	destination state
s1	o2o1o2o2	s1
s2	o2o2o1o2	s2
s3	o2o2o2o1	s3
s4	o1o2o2o1	s3
s5	o1o2o2o2	s5

このような考え方の仕様である．Naitoらの生成方法は，乱数を発生させて状態遷移を起こす入力系列をランダムに生成していく．そして，ある時点で生成した系列がすでに前に存在していたら，その系列を捨てる．この手順を繰り返してすべての欄を通過する系列を求めていく方法をとっている．

この生成手法による試験系列では，状態遷移表の出力関数の誤りは検出することが可能であるが，状態遷移関数に誤りがある場合に，検出できない場合があることが知られている．たとえば，状態遷移機械Mに対してその実現されたシステムM1が**表-2**のように表現されるとき，トランジションツアーとして（i1i2i1i2i1i2i2i1i2i1）がMの仕様から抽出される．ただし，この系列の初期状態は遷移先になっている数が少ないs5としている．このとき，実現システムM1の応答は，Mから想定される応答と同じ（o1o2o2o1o1o1o2o2o2o2）を返す．しかし，この実現は明らかに状態s5における状態遷移先が誤っており，この入力系列では正しく試験できていないことがわかる．

4.1.2　判定系列を使う方法

状態遷移関数の検査を厳密に行うには，入力を与えて状態遷移を行わせた後で，その遷移先の状態が正しく仕様で規定された状態に遷移していることが確認できなければならない．そのような状態を検査するための特別な入力系列として判定系列DSが知られている．

判定系列DSは，状態遷移表から判定木を生成して，その枝をたどることによって求める．Mの判定系列DSは（i1i1i1i1）のようになり，判定系列DSを与えたときの出力系列を**表-3**に示す．この表に示すように，判定系列DSを入力する前の状態に応じた出力系列が得られることがわかる．この結果を使って，遷移先の状態が正しいかどうかを判定することができる．

判定系列法は，上記のような特徴をもつ判定系列を用いて状態遷移関数の検査を行う試験系列を生成するものである[7]．判定系列法では，（I）状態の存在を確認する系列，（II）規定された入力を与えた後に規定された遷移先に遷移しているかどうかを確認する系列，の2つの系列を生成し，順次適用するものである．

（I）の系列では，各状態にDSを2回ずつ与える．1回目で状態の存在を確認し，2回目でDSを与えた後の状態遷移が正しく行われたことを確認する．また，（I）の系列において，DSを与えた後の状態が確認されているので，（II）では1つの入力，DSの後に続いてさらに入力DSという順序に系列を連結できるのが特徴である．

この手法で生成された系列は，遷移関数，出力関数のすべての誤りを検出できる．DS法による検証系列生成に要する計算量は，DSを生成するために状態数をmとすると最高（m-1）m^mのオーダの計算を必要とする．また，試験の対象となる状態まで遷移させるための系列（リセット系列）の生成に要する計算量は，有向グラフの最短路を求めるDijkstra法を使えば，1つの状態あたりm^2のオーダが必要であり，全状態についてはm^3のオーダである．

判定系列は，すべての状態遷移機械に存在することは保証されていない．これは，この手法を適用するときの問題となる．

4.1.3　その他の状態判定用系列

判定系列DS以外に，状態を検査するのに利用できる系列に以下のものがある．

- ユニークIO系列UIO[8]
- 特性系列集合W[9]

UIOは，判定系列と同様に1つの系列によって状態を識別することができる．しかし，判定系列と異なるのは，状態ごとに判定のための系列が異なることがある点である．判定系列やUIOが，すべての有限状態機械で存在することが保証されていないのに対して，特性系列集合Wは冗長な状態を含まない有限状態機械であれば存在することが保証されている．しかし，Wは複数の系列を用いることによって1つの状態を判定するためやや複雑となる．

4.2　イベント順序論理に基づく導出方法

イベント順序論理に基づく仕様記述言語にLOTOSがある．LOTOSでは，その意味モデルをラベル付き遷移システム（Labeled Transition System: LTS）によって定義する．LTSは，次のように定義される．

Sys＝<S, A, T, s0>
S ：状態の集合
A：アクションの集合
T ：遷移の集合（T⊆S×A×S）
s0（Sの要素）：Sysの初期状態

FSMに基づく試験と，LTSに基づく試験の違いは，前提とする仕様に対する制約である．FSMでは，仕様の完全定義や決定性，試験対象の状態数が有限などの制約があったが，LTSに基づく試験ではそのような仮定をおかない．

すなわち，LTSに基づく試験ではシステムの仕様で規定された初期状態から各終端の状態までの試験系列をすべて実行することになる．したがって，試験仕様はLTSのイベントツリーで表現される．

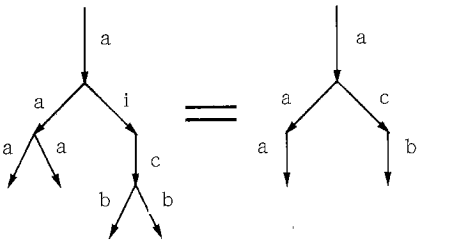
LTSで記述されたシステムの仕様は，一般に繰り返しで表現される無限の動作を表現できるが，これを無限に試験することは現実には不可能であるので，ある上限をもって打ち切る手段が必要である．また，変数を含むLTSでは，イベントの発生時に無限の種類のイベントを許すこともあり，これをすべてのパスに渡って試験することも不可能である．これについても，ある代表値を用いるなどの手段によって削減することが必要となる．

(1) 模倣性等価と試験系列

LTSでは，2つのシステムが等価であると判定する基準として，双模倣等価という概念を用いる．双模倣等価には，強双模倣等価と弱双模倣等価がある．それぞれ，次のように定義される．強模倣等価とは，1つのシステムであるアクションが発生して状態が変化するとき，もう1つのシステムでも同じアクションが発生して状態が変化するための対応する状態が存在するという等価性である．この等価性では，アクションの実行順序の観点で互いに区別できない．

一方，弱双模倣等価とは，1つのシステムで，1つのアクションと0個以上の観測できない内部アクションが発生して状態が変化するとき，もう1つのシステムでも同じアクションと0個以上の内部アクションが起こって状態が変化するための対応する状態が存在することである．弱双模倣等価のほうが，等価性の条件としては緩やかな条件となる．

たとえば，**図-4**（a）は通信システムの仕様をイベントツリーで表現したものである．iは内部イベントを示し，外部からは観測できない．これに対する双模倣等価で簡約化された仕様は**図-4**（b）のようになり，これを試験仕様として使うことができる．文献10）では，弱模倣等価性によりシステム仕様を変形して，イベントツリーに合流のない最小の木を生成するアルゴリズムを提案している．この木をたどることで，試験系列を求めるものである．



(a)システム仕様　　(b)弱双模倣等価な試験仕様

図-4 LTSの弱双模倣等価性

(2) テストパラメタの決定手法

LTSに変数を扱う機能が拡張されている言語では，イベントツリーの分岐の条件や，イベントの発火の条件が加わる．したがって，イベントツリーをたどるだけでは試験系列の候補を抽出するだけとなり，その分岐なり発火を可能とするための実際のパラメタ値を決定する必要が出てくる．

これに対して，文献11）では，扱う変数を整数とブール値に限定し，整数上のオペレータを加減算と大小比較に限定している．そして，上記の各条件を満足する整数値を求める問題を，線形計画問題として解いている．

また，仕様によっては到達可能性木が無限の長さになるものもある．この場合，ルートノードから深さがある一定値である到達可能性木を作成して，その葉ノードに対するテスト系列を求めることで対応している．

5. まとめ

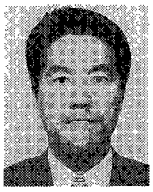
通信プログラムの試験技術について，適合性試験，相互運用性試験の概要，試験系列生成方式などについて概説した．今後は，通信プログラムがマルチメディア通信などのリアルタイム性をもつようになってきていることから，リアルタイム特性を試験する技術が重要になると思われる．形式記述技法においても，最近では時間制約を記述できるように拡張する動きが活発である．時間制約を記述した形式仕様からの，リアルタイム特性の試験用データの自動生成などが今後開発されてくるであろう．

参考文献

1) ISO: OSI-Conformance Testing Methodology and Framework Part1: General Concepts, ISO/IEC 9646-1 (1991).
2) ISO: OSI-LOTOS-A Formal Description Technique Based on the Temporal Ordering of Observational Behavior, ISO 8807 (1989).
3) ISO: OSI-Estelle-A Formal Description Technique Based on

an Extended State Transition Model, ISO 9074 (1989).
4) Takahashi, K. et al.: Design and Implementation of an Interoperability Testing System－AICTS, Proc. of IFIP 7th Workshop of Protocol Test Systems, pp.125-140 (1995).
5) ISO: Framework: Formal Methods in Conformance Testing, ISO/IEC DIS 1324-1 (1997).
6) Naito, S. and Tsunoyama, M.: Fault Detection for Sequential Machines by Transition-Tour, Proceedings of IEEE Computing Conference, pp.238-243 (1981).
7) Gonenc, G.: A Method for Design of Fault Detection Experiment, IEEE Transactions on Computers, Vol.C-19, No.6, pp.551-558 (1970).
8) Aho, A. V., Dahbura, A. T., Lee, D. and Uyar, M. U.: An Optimization Technique for Protocol Conformance Test Generation Based on UIO Sequences and Rural Chinese Postman Tours, Protocol Specification, Testing and Verification VII, pp.75-86 (1988).
9) 佐藤, 宗森, 井手口, 水野: 有限オートマトンに基づくシステムの試験系列自動生成法の提案－単一遷移検査系列法－電子情報通信学会論文誌, Vol.J72-B-I, No.3, pp.183-192 (1989).
10) 岡崎, 高橋, 白鳥, 野口: LOTOS仕様からの効率的な試験系列の自動生成法, 電子情報通信学会論文誌, Vol.J74-B-I, No.10, pp.733-747 (1991).
11) 李, 東野, 谷口: データを含むLOTOS記述に対するテスト系列の自動生成の一手法, 電子情報通信学会論文誌, Vol.J75-B-I, No.11, pp.734-743 (1992).

(平成9年10月2日受付)



佐藤　文明 (正会員)

昭和37年生．昭和61年東北大学大学院工学研究科電気及通信工学専攻博士前期課程修了．同年三菱電機（株）入社．通信ソフトウェアの研究開発に従事．平成7年1月より静岡大学工学部助教授．現在，同大学情報学部助教授．通信ソフトウェア，形式言語，分散処理システムに関する研究に興味をもつ．電子情報通信学会，IEEE Computer Society 各会員．工学博士．e-mail:sato@cs.inf.shizuoka.ac.jp