マルチメディア通信と分散処理　49－3
（１９９１．３．７）

形式記述技法を用いた適合性試験システム

勝山　光太郎　　佐藤　文明　　水野　忠則

三菱電機株式会社　情報電子研究所

ＩＳＯおよびＣＣＩＴＴでは，ＯＳＩのためのプロトコルを規定するだけでなく，プロトコル仕様を曖昧性なく記述するための形式記述技法の標準化，試験の方法論や抽象試験仕様の標準化を行っている．形式記述技法を用いた適合性試験については，ＩＳＯで新しくプロジェクトが開始された．本報告では，適合性試験の方法と枠組みに則った試験システムＦＯＲＥＳＴについて述べる．ＦＯＲＥＳＴは，ＳＤＬによるプロトコル仕様記述から自動的に，ＴＴＣＮで書かれた試験目的に応じた試験シーケンスを生成することを特徴とし，試験仕様書の作成から，試験の実行，試験成績書の作成を一貫して支援する．

A Conformance Testing System using Formal Description Technique

Kotaro Katsuyama　Fumiaki Sato　Tadanori Mizuno
Information Systems and Electronics Development Laboratory
MITSUBISHI ELECTRIC CORPORATION
5-1-1 OFUNA KAMAKURA-CITY KANAGAWA 247 JAPAN

Protocols for OSI, Formal Description Techniques and Conformance Testing Methodology and Frameworks have been standardized in ISO and CCITT. A project for conformance testing using formal approach is just started in ISO/IEC JTC1/SC21/WG1. This paper describes the design of a testing environment for communication software, called FOREST. In FOREST test sequences for each test purpose are generated automatically from SDL specification, and the testing process from test specification to test execution is supported systematically.

1．はじめに

異種システム間の相互接続を目的とし，ＩＳＯおよびＣＣＩＴＴでは，ＯＳＩ（開放型システム間相互接続）の検討を進め，各種プロトコルを標準化してきた．これらの，ＯＳＩプロトコルの標準化と並行して，プロトコルを厳密に記述する目的で，形式記述技法に基づく仕様記述言語の標準化も進められており，ＳＤＬ（Specification and Description Language）[1]，ＬＯＴＯＳ（Language of Temporal Ordering Specification）[2]，Ｅｓｔｅｌｌｅ（Extended State Transition Language）[3]といった言語が国際標準として規定されている．さらに，プロトコルのデータ単位を規定するために，構文規定言語として，ＡＳＮ．１（Abstract Syntax Notation 1）が規定されている[4,5]．

また，開発されたソフトウェアがプロトコル規約に正しく適合しているかを試験する適合性試験技術も重要となってきている．ＯＳＩプロトコルに関する適合性試験の標準化は，ＩＳＯを中心に行われている．そこでは，適合性試験の方法と枠組みを定め，それに基づき，各プロトコル毎に抽象試験仕様を標準化している．また，抽象試験仕様を記述するための試験仕様記述言語として，ＴＴＣＮ（Tree and Tabular Combined Notation）を標準化している[6]．

さらに，実際の通信システムの開発には，試験コスト及び試験期間の削減が重要な問題になっている．最初に，試験の目的と試験の手順が決定され，つぎにその目的にあった試験が通信システムの仕様から生成される必要がある．現在その試験設計の作業は専門家によって行われている．しかし，その作業は手作業であるため，試験の費用，正確さ，試験漏れ，試験仕様の開発時間といった点で多くの課題が存在する．

通信システムの試験には，適合性試験，性能試験，堅牢性試験などがある．また，適合性試験のなかにも，基本接続試験，機能試験，動作試験，適合性解決試験等の試験が含まれている．これらの試験は，適用される目的も，適用される時期も異なる．試験設計者は，これらの各試験段階に適した試験を設計することが望まれている．

こうした背景から，試験シーケンスや試験データの自動生成アルゴリズムに関する研究や，形式記述技法を用いた試験システムに関する研究が行われてきた[7,8]．ＩＳＯでも，適合性試験の形式的アプローチというプロジェクトが１９９０年５月からスタートしているが，現状は色々な方式が列挙されるにとどまっている[9]．

このため，我々は，形式的記述技法に基づく標準化された仕様記述言語を利用した体系的な試験環境ＦＯＲＥＳＴ（FORmal Environment for Systematic Testing）を提案し，上で述べたような段階的試験に適した試験シーケンスの自動生成を実現し，さらに試験仕様書の開発から試験実行，試験成績書の作成までを一貫して支援する試験システムを実現した．

以下，２章では，仕様記述言語と適合性試験について概観し，３章では，設計方針を述べ，４章では，ＦＯＲＥＳＴのソフトウェア構成および機能について述べる．５章では，ＦＯＲＥＳＴを，ＯＳＩ応用層のプロトコルの１つであるＣＣＲ[10]のソフトウェアの試験に，一部適用し評価した結果を報告する．

2．仕様記述言語と適合性試験手法

2．1　仕様記述言語

形式記述技法に基づく仕様記述言語としては，ＳＤＬ，ＬＯＴＯＳ，Ｅｓｔｅｌｌｅがある．ＳＤＬは，ＣＣＩＴＴにおいて交換機の仕様を規定するために開発された言語で，拡張有限状態遷移機械の概念をもとに，フローチャートに似たグラフ表現形式およびそれと同等の意味をもつテキスト表現形式をもつ仕様記述言語である．

ＥｓｔｅｌｌｅとＬＯＴＯＳは，ともにＩＳＯにおけるＯＳＩのプロトコルを規定するために開発された．Ｅｓｔｅｌｌｅは，ＩＳＯで既に標準化が済んでいるＰＡＳＣＡＬの言語機能に対して，状態遷移機械の概念を拡張したものである．一方，ＬＯＴＯＳは，時間順序の概念をもとに，システム仕様を外部の観察点からイベントの発生順序にしたがって記述するものである．

プロトコルの構文規定言語としては，ＡＳＮ．１が標準化されている．ＡＳＮ．１では，プロトコルのデータ構造と形式を，木構造の形で抽象的に定義することによって，実際の計算機や端末上の形式に依存しないで記述できる．

試験仕様記述言語としては，ＴＴＣＮが標準化されている．ＴＴＣＮは，試験シーケンスが木構造をとることと，表を用いた表現形式をとることを特徴としている．試験仕様の規定では，試験シーケンスの規定にはＴＴＣＮを，データ形式の規定にはＡＳＮ．１を利用する方向になっている．

2．2　適合性試験

適合性試験は，ＩＳＯを中心に，その方法と枠組みが検討されている[6]．

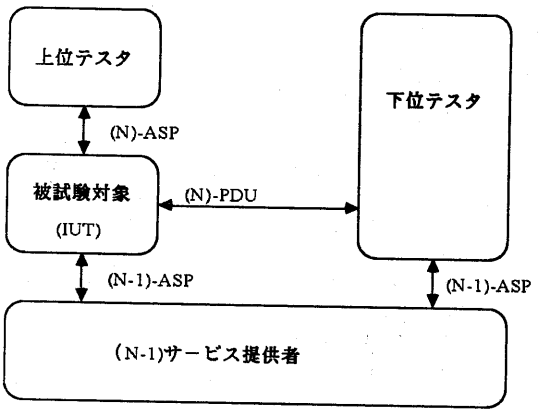
適合性試験の方法は，内部試験法と外部試験法とに大別される．内部試験法では，同一システム内に試験プログラムと被試験プログラムが存在す

る．外部試験法では，試験システムと被試験システムが通信媒体を介して接続される．

適合性試験の論理構成を図１に示す．

試験は，上位テスタと下位テスタから被試験対象（ＩＵＴ）に対して，ＡＳＰ（Abstract Service Primitive）をテストイベントとして与え，上位テスタで被試験対象からＡＳＰを，下位テスタで＜Ｎ－１＞サービス提供者からのＡＳＰおよびそれに含まれる＜Ｎ＞プロトコルデータ単位（ＰＤＵ）を観察することによって，＜Ｎ＞層にある被試験対象の動作を確認するものである．

試験の仕様記述はＴＴＣＮを利用することが規格では決められている．また，ＩＳＯでは，プロトコル毎にＴＴＣＮを用いて抽象試験仕様を決めている．



PDU : Protocol Data Unit
ASP : Abstract service primitive

図１　適合性試験の論理構成

２．３　仕様記述言語の適合性試験への適用

２．１で述べた仕様記述言語のうち，形式的記述技法に基づく仕様記述言語を用いた適合性試験のやり方には大きくわけて次の２つの方法が考えられる．

（１）試験シーケンス生成方式

形式記述技法に基づく仕様記述言語で書いたプロトコル記述から，試験シーケンスを導出する方法である．有限状態機械モデルに基づくものとして，ＴＴ法[11]，ＵＩＯ法[12]，ＳＷ法[13]といった生成アルゴリズムが存在する．

（２）直接試験実行方式

形式記述技法に基づく仕様記述言語により，テスタの状態遷移を記述し，実際にそれを動作させ，試験を行なう方法である．Ｅｓｔｅｌｌｅを利用したJ.R.Linnらの試験システムがこの方法を採用している[14]．

（１）の方法には，試験シーケンスが膨大となる欠点があるが，事前に導出するため，試験の内容がわかるという長所がある．

（２）の方法には，仕様記述と試験システムが一体化しているという長所はあるが，実行時の試験シーケンスをあらかじめ見ることができず，試験仕様書を要求されるような場合には利用できない欠点がある．

適合性試験の方法論に則ると（１）の方法が，既存の試験システムへの影響が少ないため有力と考えられるが，試験の目的に沿った試験シーケンスの導出という点を解決する必要がある．

３．設計方針

ＦＯＲＥＳＴを実現するにあたり，次のような設計方針をたてた．

（１）標準化された仕様記述言語を用いるとともに，適合性試験の方法論に則った試験システムとする．試験対象の動作仕様を記述する言語として，標準化されている言語としてはＳＤＬ，ＬＯＴＯＳ，Ｅｓｔｅｌｌｅがあるが，通信システムの技術者が馴染んでいる拡張有限状態機械モデルであり，かつ現在広く使われていることからＳＤＬを採用した．また，データタイプ定義では，ＡＳＮ．１による定義方法と，ＳＤＬやＬＯＴＯＳで使われている抽象データタイプによる定義方法があるが，抽象データタイプの定義は厳密であるが理解性が低く，計算機処理もＡＳＮ．１より困難であることから，ＡＳＮ．１による定義を採用した．また，試験仕様の文書はＴＴＣＮによる出力とした．

（２）製品検査試験の一環として，この環境のもとで試験を行うため，試験成績書が文書として残ることが必要である．このため，試験シーケンスを予め生成しておく試験方法をとることにした．また，試験の効率化を目的とし，一台のワークステーション上で試験に於ける試験仕様書の作成，試験の実行，及び試験成績書の作成を一貫して支援する．

（３）試験仕様の作成では，標準化された試験の方法論に沿った作成を支援する．そのために，段階的試験に対応した試験シーケンスを生成する．

４．ＦＯＲＥＳＴの構成

４．１　操作手順

ＦＯＲＥＳＴのオペレーションフローを以下に示す．図２中の番号と対応している．

（１）ＳＤＬ／ＰＲの記述

状態遷移の仕様をＳＤＬ／ＰＲファイルに定義する．この作業は，エディタを用いて行われる．

（２）試験シーケンスの生成（ＴＥＮＴ）

試験シーケンスが（１）で記述されたＳＤＬ／ＰＲファイルから生成される．

（３）試験仕様記述（ＴＥＳＰＥＣ）

試験仕様は，（２）で生成された試験シーケンスを参照し，ＴＴＣＮに定義されているような表形式に基づいたユーザインタフェースで記述される．

（４）ＡＳＮ．１の使用

ＯＳＩ応用層のプロトコルデータ単位はＡＳＮ．１で定義されている．システムでは，プロトコルデータ単位の与えられたＡＳＮ．１記述を用いる．

（５）試験データ生成（ＡＰＲＩＣＯＴ）

試験データは，ＡＳＮ．１で定義された仕様から半自動的に生成される．

（６）試験実行環境（ＴＥＸＥＣ）

試験は，（２）で生成された試験シーケンスと，（５）で生成された試験データを用いて実行される．

（７）試験成績書記述（ＴＥＳＰＥＣ）

試験成績書は，試験仕様と試験結果から自動的に生成される．試験成績書の概要も出力される．この概要には，トータルテストケース，合格テストケース，不合格テストケース等の統計情報が含まれる．

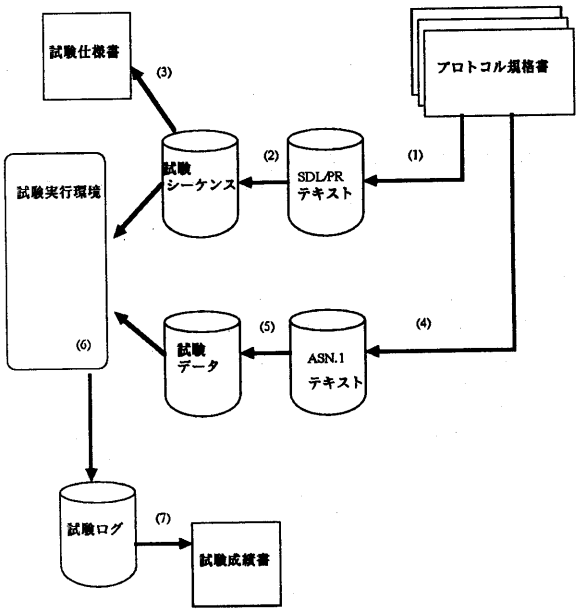


図２　ＦＯＲＥＳＴの試験手順

ＦＯＲＥＳＴは，以下の４つのサブシステムを１つのワークステーション上に統合したシステムとなっている．

４．２　ＴＥＮＴ

ＴＥＮＴは，仕様記述言語ＳＤＬから試験シーケンスを生成するシステムである．ＴＥＮＴは，様々な試験段階と試験項目に応じた試験シーケンスを生成する必要がある．

そのためＴＥＮＴでは，複数の生成手法を実現し，それぞれの生成手法に対して，分割生成機能を拡張した．

（１）ＭＴ（Modified TT）法－ＴＴ法が，全ての状態に規定された遷移を全て通るようなシーケンスを求める手法であるのに対して，基本接続試験用の簡潔で，ある網羅性を持つシーケンスを生成する．つまり，ＭＴ法では，全ての状態遷移を対象とするのではなく，全ての状態に到達できることを確認するシーケンスを生成する．ＭＴ法は，全ての状態を一通り通過するような，なるべく短いシーケンスを求める方法である．

（２）ＴＴ（Transition Tour）法－ＴＴ法は，現在広く用いられている生成方法であり，標準的な試験シーケンス生成手法であると考えられる．状態遷移表のすべての欄を，なるべく短い手順で網羅するような試験シーケンスを生成する．但し，試験の範囲に限定した試験シーケンスを得るために，状態遷移表の他に，フェーズ情報や機能分割の情報をシーケンス生成時に参照するように拡張されている．

（３）ＳＷ（Single transition checking method using W set）法－ＳＷ法は，ＣＳ法の生成アルゴリズムに対して，判定シーケンスＤＳの代わりに，特性集合Ｗを状態の判定のために用いた試験シーケンス生成手法である．ＳＷ法で生成された試験シーケンスは，状態を確認しながら試験できるので，出力関数と状態遷移関数の実現の誤りをともに検出できる．ＳＷ法は，ＭＴ法，ＴＴ法と比べて，シーケンス長は長くなる．

（４）フェーズ分割機能－通信システムの状態はフェーズごとにいくつかのサブセットに分割される．図３の例に示すように，それぞれの状態のサブセットについて試験シーケンスを生成することによって，試験対象以外のフェーズを通過することの少ない試験シーケンスを求めることができる．例えば，ＴＴ法やＳＷ法では，状態遷移の単位で試験シーケンスの生成を行なうため，状態遷移が他のフェーズに出てしまう場合，そこで一旦試験シーケンスを打ち切って，改めてそのフェーズの別の状態から開始するようなシーケンスを求める

ことにする．

（５）機能分割機能－一つのフェーズ内においても，正常処理，例外処理，異常処理やその他の目的別の処理が含まれている．これらの処理を対象とした試験シーケンスの生成を行なう方法としては，それぞれの目的に応じたフラグを状態遷移に付けておくことにする（図４）．ＴＴ法やＳＷ法では，シーケンス生成時にそのフラグを基準として対象とする状態遷移を識別することによって，目的別の試験シーケンスを生成できるようになる．

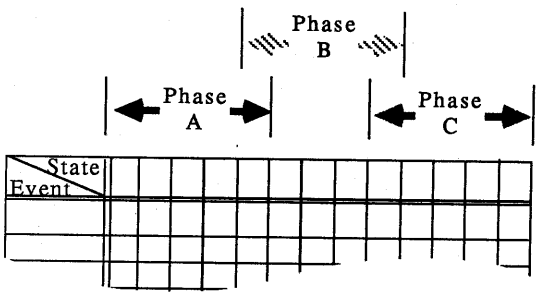
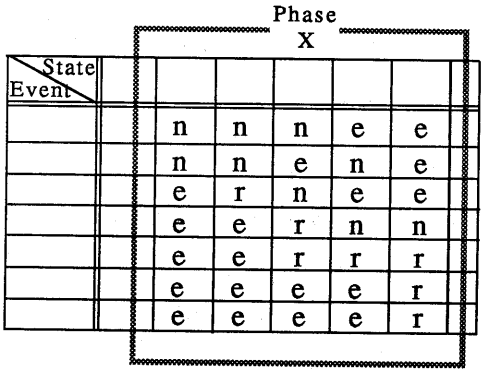


図３　フェーズ分割の例



n: 正常処理機能 e: 異常処理機能 r: リカバリ機能

図４　機能分割の例

ＴＥＮＴは，４つのツール群とユーザインタフェース部分で構成されている．ＴＥＮＴを構成するツールには，ＣＣＩＴＴ（国際電信電話諮問委員会）が勧告する仕様記述言語ＳＤＬのテキスト表記から，ＦＳＭの状態遷移表を生成するＳＤＬ・状態遷移表トランスレータ（以下単にトランスレータ），状態遷移表を編集したり，機能分割情報を入力する状態遷移表エディタ，フェーズ情報を入力するフェーズ情報エディタ，そして状態遷移表から実際に試験シーケンスを生成する試験系列（以下単にジェネレータ）がある．ＴＥＮＴの内部構成を図５に示す．

試験仕様の開発者（ユーザ）は，ＳＤＬ用のグラフィックエディタ[15]や，テキストエディタ等によって作成されたＳＤＬ仕様から，トランスレータを使って状態遷移表に変換する．状態遷移表のデータに対して，ユーザは試験対象とする機能毎に機能分割情報を各遷移に対するフラグとして入力する．状態遷移表エディタは，機能分割情報を，入力単位，出力単位，及び状態単位で入力し編集することができる．更に，試験の範囲を限定するために，試験のフェーズ情報を入力する．状態遷移表と，機能分割情報及びフェーズ情報から，ジェネレータによって試験シーケンスを生成する．ジェネレータは，試験の段階を選択することによって，適切な生成方式を選択し試験シーケンスを生成する．

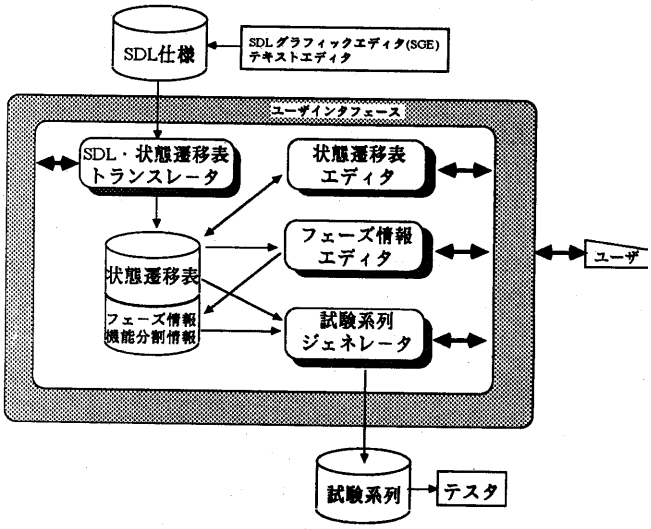


図５　ＴＥＮＴの構成

４．３　ＡＰＲＩＣＯＴ[16]

ＡＰＲＩＣＯＴは，４．１（５）の試験データ生成を実現するためのソフトウェアである．

ＡＰＲＩＣＯＴは，i）ＡＳＮ．１定義から階層構造を定義するテーブル（型定義テーブル）を出力するプリコンパイラ，ii）階層構造データを符号化・復号化するためのコーデックライブラリおよびiii）プログラムの試験時のために，ＡＳＮ．１で記述されたデータの階層関係を解析，表示するためのデバッガ，の３つの部分から構成される．

このＡＰＲＩＣＯＴを試験データの生成に利用するにあたり，次のような拡張を行った．

応用層，プレゼンテーション層，セション層のプロトコルデータ単位は，それぞれ，どのＰＤＵを利用して運ぶかが規定されている．この規則に基づき，マッピングファイルを作成する．ＡＳＮ．

１を支援するツールであるＡＰＲＩＣＯＴに，このマッピングファイルを参照してマルチプロトコルデータを生成する機能を追加した．ここでいうマルチプロトコルデータとは，応用層の複数プロトコルのデータ，プレゼンテーション層のプロトコルのデータをマッピング情報を利用し，結合したものをセションサービスの利用者データとして作成したものである．

また，ＡＳＮ．１によるプロトコルデータ単位の定義を使って，利用者が各プロトコルデータ単位のパラメタを指定できるインタフェースを提供するようにした．

４．４　ＴＥＳＰＥＣ

ＴＥＳＰＥＣは，試験仕様書及び試験成績書を作成するためのユーザインタフェースを提供する．その基本的な考え方は，試験の仕様を記述しながら，試験データや試験シーケンスを並行して作成していくことと，試験仕様書の入力形式を，ＩＳＯが規定する適合性試験の方法論と枠組みに基づいて行うことである．

はじめに，試験仕様の全体概要を入力し，次に，試験に用いるＡＳＰやＰＤＵの選択を行う．試験に用いるタイマや上位テスタ，下位テスタに関する記述を行う．次に，試験スイートのグループやケースに関する記述を行う．

ＡＳＰやＰＤＵの定義の部分で入力されたパラメータは，ＡＰＲＩＣＯＴの入力として用いられ，試験データが仕様書を作成すると同時に準備される．また，試験グループや試験ケースの入力の段階で，試験の対象範囲を限定する情報を入力する．その情報を基に，ＴＥＮＴが試験シーケンスを生成し，その試験シーケンスが試験仕様書の中に取り込まれる．

４．５　ＴＥＸＥＣ（試験実行環境）

試験実行環境では，一台のＷＳで通信ソフトウェアの設計から制作，試験までを一貫して支援することを目的としているため，内部試験法を採用する．但し，通信路シミュレータと実際の通信路と置き換えることにより，外部試験法でも実施可能となるようにした．

図６に試験実行環境の構成を示す．

セションシミュレータ[17]は，セション層のサービスを模擬するものである．

被試験対象プログラムは，プレゼンテーション層と応用層のプロトコルを実装したものである．上位テスタは，被試験対象と上位層とのイベントを観測制御するためのソフトウェアである．試験シーケンスファイルは，ＴＥＮＴが生成したものを利用する．試験データファイルは，ＡＰＲＩＣＯＴを利用して作成されたプロトコルデータである．試験ログファイルは，下位テスタが試験実行の履歴を出力するものである．ＴＥＳＰＥＣがこれを参照し，試験成績書を作成する．

試験協調の手順を簡素化するために，フェリー法[18]を採用した．

われわれは，フェリー制御チャネルとして，コネクションレスセションサービスを利用して，簡単なフェリー制御プロトコルを開発した．

試験の実行は，セションシミュレータ，被試験対象プログラム，上位テスタ，及び下位テスタをこの順序に従って起動することによって開始される．もし，試験が失敗した場合は，試験の実行は中断され，試験ログにより不具合の解析を行なう．

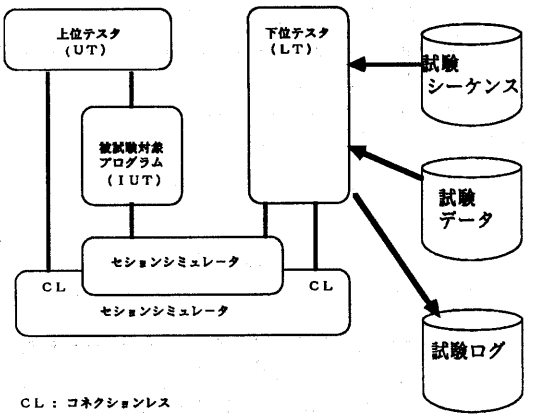


図６　試験実行環境の構成

５．　ＦＯＲＥＳＴの適用

ＦＯＲＥＳＴをＯＳＩ応用層プロトコルの１つであるＣＣＲプロトコルの開発に適用した[19]．

ＣＣＲは，分散処理全体の処理の同期をとるためのプロトコルであり，本来分散環境で使用されるため，複数のエンティティ間でのやり取りがあるが，今回は１対１通信に関する適合性試験を行うこととした．

試験実行環境の観点では，フェリー法を採用することによって，上位テスタのつくりが単純になるため，外部試験法に移行した場合，被試験システムの上位テスタの作成が容易となる．また，１台のワークステーション上で試験仕様書の作成から，試験の実行，成績書の作成までを一貫して支援することが可能となり，試験の効率があがった．

試験実行環境の課題としては，次の点があげられる．

(1)試験データの確認を単純化のためバイナリ比較としたため，プロトコルデータ単位のリザーブエ

リアや，本来関係のないフィールドも値が合致している必要があり，試験に若干手間取った．
(2)テスタが複数のコネクションを制御できないため，１本のアソシエーションに付いてしか試験できなかった．

試験シーケンス生成の観点からは，以下の事があげられる．

（１）ＣＣＲプロトコルで生成された試験シーケンスの長さは，

基本接続試験：　１１１イベント
動作試験Ａ（網羅）：　２４０１イベント
動作試験Ｂ（徹底）：１２３２１６イベント

であった．

（２）実際の試験に用いる試験シーケンスを作成するために，ＣＣＲプロトコルの仕様だけでなく，ＡＣＳＥのアソシエーション確立の手順も記述する必要があった．つまり，ＣＣＲの仕様では，障害が発生した場合に，障害発生を通知して次の状態に写ることになっているが，このとき同時にアソシエーションも再確立しなければならない．このアソシエーション再確立部分はＣＣＲの仕様に追加する形で試験シーケンス生成用のＳＤＬ／ＰＲ記述に加えた．

（３）ＳＤＬ／ＰＲから内部遷移表へ変換する過程で，条件分岐の部分は，条件を満たすイベントを入力するという仮定で分割している．しかし，ＣＣＲでは，外部から入力イベントで制御できないコミットメント状態変数という条件要素が存在している．この変数を参照している条件分岐では，同じ入力イベントでも異なる動作をすることから，結果が予測できず，試験手順が生成できない．そのため，今回の試験シーケンスの生成からは除いた．

（４）シーケンスの出力形式はＴＴＣＮの動作部形式としてあるので，テスタはこれをそのまま読み込んで試験に用いることができた．出力データには，被試験システムが遷移しているはずの状態やシーケンスの性質（試験用のシーケンス，試験対象の状態へ遷移させるためのシーケンス等）をコメントとして記述している．このデータによって，エラーが発生したときの原因の解析に役だった．

（５）試験シーケンスの分割生成において，試験シーケンスの本数の増加に伴い，試験対象とする状態まで状態遷移をさせる予備的なシーケンスが増加するため，全体のシーケンス長が増加する．今回の実験では，生成手法を単純に実現した場合の生成結果と比べて，それほど多くの増加はなかった．ＴＥＮＴの動作試験Ａのシーケンスの総和は，ＴＴ法の分割生成機能が無い場合に対して，最大２５％の増加であった．分割数とシーケンス長の増加率との関係については今後の実験で明らかにする予定である．

また，ＴＥＮＴでは，試験の効率化のために，試験シーケンスを分割することを提案している．それは，試験シーケンスの総和が短くても，１本の長いシーケンスを生成した場合に非効率的な試験となるからである．

試験の期間に発見された誤りにおいては，仕様自体の記入が不足しているもの，仕様に誤りがあるものが十数件発見された．これは，ＣＣＲの仕様自体がまだ完成されたものではなかったことを示している．ＴＥＮＴでは，ＳＤＬ／ＰＲか変換された内部遷移表に基づき，試験シーケンスを生成するが，この生成の過程において，状態の到達性の検査を行っている．つまり，入力された仕様自体の誤りについての検証を行う．これによって，上記のような製品試験以前の誤りが検出され，試験の効率化に十分役だったと考えられる．

試験データ生成の観点からは，ＰＤＵのパラメータをセットするインタフェースを用意したことから，様々な値の試験が可能となり，有効だった．

また，試験データのＰＤＵのＡＳＮ．１定義は，仕様を直接参照することが可能であり，パラメータフィールドの設計等の労力が削減できた．

今回のＣＣＲの試験において，実際に出力された試験仕様書は，１３６頁にわたっており，そのうち，約５０％が試験仕様の動作部で，このシステムによって自動生成されている．また，のこりの５０％の部分についても，テーブルの表題や，各章，節の表題については自動生成されている．

６．　おわりに

国際標準に準拠した仕様記述言語の検証のために，仕様記述言語を利用した体系的な試験環境ＦＯＲＥＳＴを提案し実現した．

ＦＯＲＥＳＴを利用することによって，試験仕様書の作成，試験の実行，試験成績書の作成を一貫して行うことができ，ＯＳＩ高位層ソフトウェアの試験段階の効率をあげることが可能となった．特に，ＦＯＲＥＳＴでは，仕様記述言語から試験シーケンスを自動生成することを特徴としており，網羅性の高い試験が可能となった．試験シーケンスの自動生成による省力化もさることながら，生成過程での仕様の誤りチェックに有効であり，仕様記述言語を利用したことの有効性が確認できた．

今回はＳＤＬを適用したが，同種の言語であるＥｓｔｅｌｌｅは，ほぼ同一の方法で実施可能と考えられる．ＬＯＴＯＳに関しては，言語形態が異なるために，今後の課題となる．

また，実際の通信システムへの適用に関しては，ＩＳＯのプロトコル規格が，単一のコネクション通信の場合を想定しているために，１対１通信についての適合性試験を行った．今後は，複数コネクション通信の場合の適合性試験に対して，仕様記述をどのように展開するのかが課題となる．

また，ＩＳＯで検討が開始された適合性試験における形式的アプローチのプロジェクトにおいても，既存の試験方法論や試験システムからの移行を考慮すると我々の提案する段階的試験シーケンスの生成においてＴＴＣＮを出力する方式は，十分に利用可能と考えられる．

＜参考文献＞

［１］CCITT:Functional Specification and Description Language, Recommendation Z.100-Z.104(1985).

［２］ISO:Information Processing Systems - Open Systems Interconnection - A Formal Description Technique Based on the Temporal Ordering of Observational Behavior" ISO 8807(1989).

［３］ISO:Information Processing Systems - Open Systems Interconnection - Estelle - A Formal Description Technique Based on an Extended State Transition Model, ISO/DIS 9074(1987).

［４］ISO:Information Processing Systems - Open Systems Interconnection - Specification of Abstract Syntax Notation One (ASN.1), ISO 8824(1988).

［５］ISO:Information Processing Systems - Open Systems Interconnection - Specification of Encoding Rules for Abstact Syntax Notation ONE (ASN.1), ISO 8825(1988).

［６］ISO:Information Processing Systems - OSI Conformance Testing Methodology and Framework, DIS9646(1990).

［７］v.Bochmann,G.:Usage of Protocol Development Tools: The Results of a Survey;Protocol Specification, Testing and Verification VII An International Symposium, pp147-170(1987).

［８］R.L.Probert, H.Ural, and M.W.A.Hornbeck:An Integrated Software Environment for Developing & Validating Standardized Conformance Tests, IFIP Protocol Specification, Testing and Verification VIII, pp87-98(1988).

［９］ISO:Working Draft for multi-party test methods, ISO/IEC JTC1/SC21 N5076(1990).

［１０］ISO:Information Processing Systems - Commitment, Concurency and Recovery, DIS9804-9805/2(1987).

［１１］Naito, S. and Tsunoyama, M.: Fault Detection for Sequential Machines by Transition Tours, Proc. of IEEE Computing Conference, pp.238-243(1981).

［１２］Sabnani, K., Dahbura, A.:A protocol test generation procedure, Computer Networks and ISDN Systems 15, pp.285-297(1988).

［１３］佐藤文明，宗森純，井手口哲夫，水野忠則：有限オートマトンに基づくシステムの試験シーケンス自動生成法の提案ー単一遷移検査系列法ー，信学会論文誌，Vol.J72-B-I, No.3, pp.183-192(1989).

［１４］A.Iwabuchi,R.J.Linn,and J.P.Favreau: APPLICATION OF FORMAL SPECIFICATION TECHNIQUES TO THE SPECIFICATION OF THE MHS SYSTEM, Proceedings of the second international symposium on interoperable information systems, pp255-262(1988).

［１５］宗森純，水野忠則：ＳＤＬグラフィックエディタの設計と製作，情報処理学会論文誌，Vol.29, No.7, pp.676-685(1988).

［１６］T.Nakakawaji,K.Katsuyama,N.Miyauchi, T,Mizuno:DEVELOPMENT AND EVALUATION OF APRICOT (TOOLS FOR ABSTRACT SYNTAX NOTATION ONE), Proceedings of the second international symposium on interoperable information systems, pp55-62(1988).

［１７］勝山光太郎，中川路哲男，宮内直人，水野忠則：セションシミュレータによる通信ソフトウェアの開発，第３７回マルチメディアと分散処理研究会(1988).

［１８］G.v.Bochmann and C.S.He : Ferry approaches to protocol testing and service interfaces, Proceedings of the second international symposium on interoperable information systems, pp303-309(1988).

［１９］中川路哲男，宮内直人，勝山光太郎，水野忠則：ＯＳＩ　ＣＣＲの実現，第３７回マルチメディアと分散処理研究会(1988).

［２０］勝山光太郎，佐藤文明，宮内直人，中川路哲男，水野忠則：ＯＳＩ高位層ソフトウェア試験システム，第３９回マルチメディア通信と分散処理研究会(1988).