

## 適合性試験システム ACTS における試験スイートの開発

似内 智\* 高橋健一\* 後藤邦弘\* 石幡吉則\* 高橋 薫\*\*  
\* (株) 高度通信システム研究所 \*\* 東北大学

開放型システム間相互接続技術の普及、発展とともに、その通信システムを構成する異機種端末間の相互接続試験の必要性が増している。そこで我々は、試験の自動化と汎用化を目指したアーキテクチャをもつ相互接続試験システム AICTS の構想をまとめ、現在その開発を進めている。本報告では、AICTS を実現するための第一段階として開発した、トランSPORT層クラス 0 を試験対象とする適合性試験システム ACTS の抽象試験スイートと実行型試験スイートの仕様について述べる。

さらに、ACTS とトランSPORT層クラス 0 の模擬 IUT として開発した IUT シミュレータを用いて適合性試験を行なった結果について報告する。

### Test Suite in the Conformance Test System ACTS

Satoshi NITANAI\* Ken'ichi TAKAHASHI\* Kunihiro GOTOH\* Yoshinori ISHIHATA\*  
Kaoru TAKAHASHI\*\*

\* Advanced Intelligent Communication System Laboratories (AIC)

\*\* Tohoku University

\* 6-6-3, Minamiyoshinari, Aoba-ku, Sendai, 989-32 Japan

\*\* 2-1-1, Katahira, Aoba-ku, Sendai, 980 Japan

Interconnectability testing for a variety of terminal equipment developed in conformity with the same OSI (Open Systems Interconnection) standard is very important for the diffusion of OSI. We have been developing the interconnectability testing system AICTS. As a first step of that development, we have developed the ACTS (AIC's Conformance Test System) that can be applied to the conformance testing of the transport layer class 0 protocol. This paper describes the abstract and executable test suite specification in the ACTS. Furthermore, we report the result of the conformance test execution obtained by using an IUT simulator.

## 1.はじめに

開放型システムに対する国際標準（プロトコル仕様およびサービス定義）は、あるレベルで抽象化されたものであり、必ずしも実装に必要十分な細部規定は行なっていない。そのため同一の国際標準に従って開発された情報通信端末であっても機種ごとに細部の仕様、設計は異なっている。このことは適合性試験[1], [2]に合格しても相互接続ができない理由の一つと考えられる。そこで適合性試験を通過した端末での相互接続性の確認が必要となる。情報通信システムの大規模化・多様化にともなって異機種端末間の相互接続試験に対する認識が高まってきており、研究例もいくつか報告されている[3], [4], [5], [6]が、統一的な相互接続試験アーキテクチャは未だ確立されていない。

そこで我々は、開放型システムを構成する通信ソフトウェアを試験対象とし、試験の自動化、汎用化を目指した相互接続試験システム AICTS (Advanced Intelligent Communication System Laboratories' (AIC's) Inter-Connectability Testing System) の構想をまとめ、その開発を進めている[7]。

本報告では、AICTS を実現するための第一段階として開発したトランスポート層クラス 0 を試験対象とする適合性試験システム ACTS (AIC's Conformance Test System) の試験スイートの仕様およびACTSによりトランスポート層クラス 0 の模擬 IUT (IUT シミュレータ) に対して適合性試験を行なった評価結果について報告する。

## 2.適合性試験システム ACTS の構成

適合性試験システム ACTS の構成は、図 1 に示すように、アクティブテスター (AT : Active Tester) とパッシブテスター (PT : Passive Tester) に機能分割される。AT は、試験対象システム (SUT : System Under Test) との試験データ送受信および PT 制御の役割をもつ。PT は、試験管理プロトコル (TMP : Test Management Protocol) に従い AT の指示で動作する。なお、AT と PT はそれぞれ適合性試験における上位テスター (UT : Upper Tester) と下位テスター (LT : Lower Tester) に相当する。

制御観測点 (PCO : Point of Control and Observation) は、試験対象実装 (IUT : Implementation Under Test) の上位のサービスアクセス点 (PCO(U)) と下位のサービスアクセス点 (PCO(L)) に設け、さらに A

CTS 特有の試験管理メッセージ (TMM : Test Management Message) の試験管理制御点 (PMC : Point of Management and Control) を AT の下位 (PMC(M)) に設けた。

また、コネクションは試験データが流れる試験チャネル (TC : Test Channel) の他に TMM 用の試験管理チャネル (TMC : Test Management Channel) を設けている。この方法により使用する論理回線は増えるが、試験管理の信頼性向上などの効果が期待できる。

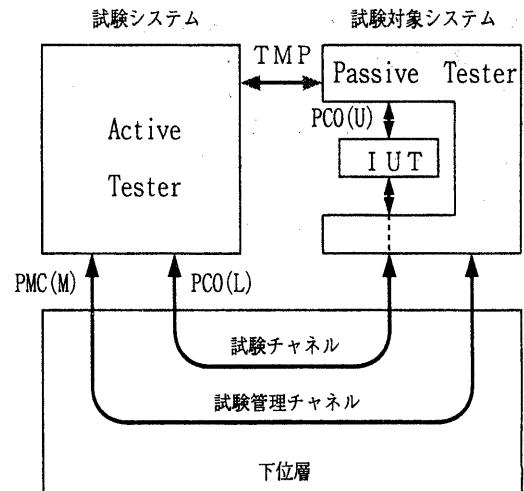


図 1 ACTS の構成

## 3.抽象試験スイート

TTCN (Tree and Tabular Combined Notation) [8] は、適合性試験の方法と枠組みに従って、抽象試験スイート (ATS : Abstract Test Suite) を記述するための言語として国際標準化されたものであり、試験に必要な試験目的、試験の条件といったドキュメント的な性質のものから、試験データの定義、試験シーケンスの定義といったものを併せて記述できるようになっている。また、TTCN にはデータの定義やシーケンスの定義に表形式を用いる TTCN.GR (Graphical form) とテキスト表現を用いる TTCN.MP (Machine processable form) の 2 つがあり、両者は同じ意味を表現することができる。

我々は、ACTS 用 ATS を TTCN.GR の表記法を用いて開発した。なお、トランスポート層クラス 0 を対象とする ATS には、OSTC (Open Systems Testing Consortium) [9]、INTAP [10] などによる

ものが知られているが、本 A T S は欧州の事実上の標準である O S T C により開発されたものと同等である。

### 3.1 試験スイートの構成

試験ケースの体系化された集合である試験スイートは、試験グループ、試験サブグループ、試験ケースの三段階の木構造からなっており、適合性試験システム A C T S の試験スイートの構成を図 2 に示す。A C T S では、試験スイートをコネクション確立、データ転送、コネクション解放の 3 つの通信フェーズに分類し、これを試験グループとし、それをさらに試験の動作に着目して、正しい動作、不正な動作、タイミングの悪い動作に分類し、これを試験サブグループとしている。そして、グループ毎に分類された試験ケース数は 40 項目であり、それぞれの試験ケースの試験目的は図 2 に示すところである。

### 3.2 試験スイート記述例

試験仕様である試験ケースには、I U T の上位および下位の制御観測点（P C O （U）およびP C O （L））と試験管理制御点 P M C （M）に対する制御・観測内容を記述し、その結果として予測される I U T の動作に対して、プロトコル仕様およびサービス定義に照らした判定を記述する。判定は、通過（p a s s），失敗（f a i l），不確定（i n c o n c l u s i v e）のいずれかである。

表 1 に、P C O およびP M C の存在場所を定義したP C O，P M C 宣言の記述例を示す。

また、図 3 には試験本体に先立つ前処理部で行なわれる A C T S 特有の試験管理チャネル（T M C）のコネクションの確立と P T 動作確認のための試験仕様記述例を示す。なお、基本的な記述法を以下に示す。

M ! A S P 名（T M M 名）：MにおいてT M M名を

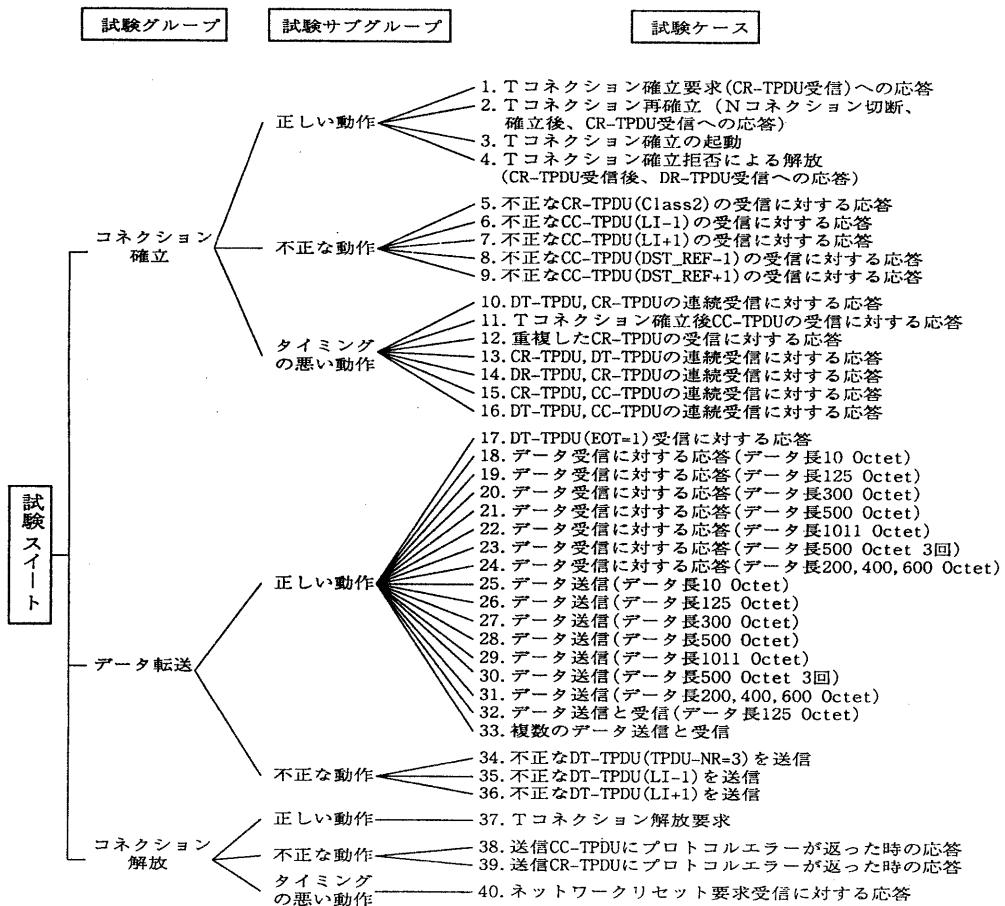


図 2 試験スイートの構成

Test Ref : TRO/TMM1/Preamble	Purpose : Test Management Channel (TMC) のコネクションの確立と PT の動作確認を行なう。				
Test ID : TMM1 / Preamble					
Defaults Ref : Def1					
Behaviour	Description	Label	Refer	Comments	Result
M ! N_CON_REQ Start (Y) M ? N_CON_CONF Cancel (Y) M ! N_DATA_REQ ( PT_ACT_CHK ) Start (Z) M ? N_DATA_IND ( PT_ACT_ACK ) Cancel (Z) ? Timeout (Z) +Pos1 / Postamble ? Timeout (Y) +Pos1 / Postamble				(1) (2) (3) (4)	

図3 試験管理チャネルTMCのコネクション確立とPT動作確認のための試験仕様記述例

データとして含むASP名を送信する。

M ? ASP名 (TMM名) : MにおいてTMM名をデータとして含むASP名を受信する。

ここでPMC (M) はMで表わし、ASPは抽象サービスプリミティブである。また'!'はATからの送信を表わし、'?'はATでの受信を表わす。図3の内容の概略をコメント欄の番号毎に以下に示す。

- (1) TMC用のネットワーク (N : Network) コネクション接続要求 (N\_CON\_REQ) をATから送信する。
- (2) TMC用のNコネクション接続確認 (N\_CON\_CONF) をATで受信する。
- (3) TMCを用いてPTの動作確認データ (PT\_ACT\_CHK) をATから送信する。
- (4) TMCを用いてPTの動作状態の通知データ (PT\_ACT\_ACK) をATで受信する。

表1 PCO, PMC宣言記述例

PCO Declaration	
NAME	Role
L	SAP at the Active Tester. Active Tester is N-1 service user.
PMC Declaration	
NAME	Role
M	SAP at the Active Tester. Active Tester is N-1 service user.

試験調和手続きのためのTMPに基づいて通信されるTMMの種類と機能を表2に示す。

図4には、試験ケース記述例（トランスポートコネクションの確立）を示す。これは図2の1番目の試験ケース、トランスポート (T : Transport) コネクションの確立要求に対するIUTの応答を見るためのものである。図4の内容の概略をコメント欄の番号毎に以下に示す。

- (1) 前処理部であり、前述の図3に示す内容により、TMCのNコネクション確立を行なう。
- (2) 前処理部であり、試験チャネル (TC) のNコネクションの確立を行なう。
- (3) ATからIUTに対してCR-TPDU (Tコネクション接続要求TPDU) を送信する。
- (4) ATがIUTからCC-TPDU (Tコネクション接続確認TPDU) を受信すれば、判定はpassとな

表2 TMMの種類と機能

TMMの種類	伝送方向	機能
PT_ACTIVATE_CHK	AT → PT	PTが動作していることを確認する。
PT_ACTIVATE_ACK	PT → AT	PTの動作状態をATに通知する。
PT_SEND_REQ	AT → PT	PTにIUTに対してASPを送信することを要求する。
PT_SEND_ACK	PT → AT	ASP送信要求に対し応答する。
PT_RECEIVE_REQ	AT → PT	PTにIUTから受信したASPをATへ送信することを要求する。
PT_RECEIVE_ACK	PT → AT	ATに受信ASPを通知する

Test Ref : TRO/0A1/B01/001	Purpose : トランスポートコネクションの確立要求を受け付けることを確認する。				
Test ID : TRO/001					
Defaults Ref : Def2					
Behaviour	Description	Label	Refer	Comments	Result
+TMM1 / Preamble +TDU1 / Preamble L ! N_DATA_REQ ( CR ) Start ( A ) L ? N_DATA_IND ( CC ) Cancel ( A ) +Pos2 / Postamble ? Timeout ( A ) +Pos2 / Postamble		CR1 CC21		(1) (2) (3) (4) (5) (6) (5)	pass fail
Comments :				(1) Test Management Channel (TMC) のネットワークコネクションの確立を行う (2) Test Channel のネットワークコネクションの確立を行う (3) AT から CR-T PDU を IUT に送信する (4) IUT から送られる CC-T PDU を AT で受信する (5) Test Channel と Test Management Channel のネットワークコネクションの解放を行う (6) タイマ値 A 以内に IUT から CC-T PDU を AT で受信しなければタイムアウト	

図4 試験ケース記述例（トランスポートコネクションの確立）

る。

(5) 後処理部であり、TMCとTC用のNコネクションの解放を行なう。

(6) ATがIUT無応答監視用のタイマAの設定時間内にIUTからCC-T PDUを受信しなければ、タイムアウトとなり、判定は fail となる。

#### 4. 実行型試験スイート

抽象試験スイート (ATS) は、試験システムによる試験実行用ではないので、ATSの抽象的な部分や省略されている部分を具現化し、試験システムが処理できる実行型試験スイート (ETS : Executable Test Suite) を作成する必要がある。そのため、ETS開発者は、AT

Sおよびプロトコル仕様を参照し、エディタによりブレーンテキスト形式のETSを作成してファイルとして保存する。試験では、テストスイートローダがそのファイルをテストドライバのメモリ上に展開し、それをテストスイートインターフェースが操作しながら各部の制御を行ない、試験を実行する。（以下、この試験実行方式を試験スイートインターフェース方式と呼ぶ。）

##### 4.1 実行型試験スイートの構成

ETSの構成を図5に示す。ETSは、プロトコル標準などで通信手順を記述する際によく用いられる状態遷移表と同様の構造を持つ状態遷移表形式実行型試験スイートである。状態遷移表の横軸は外部からの入力イベ

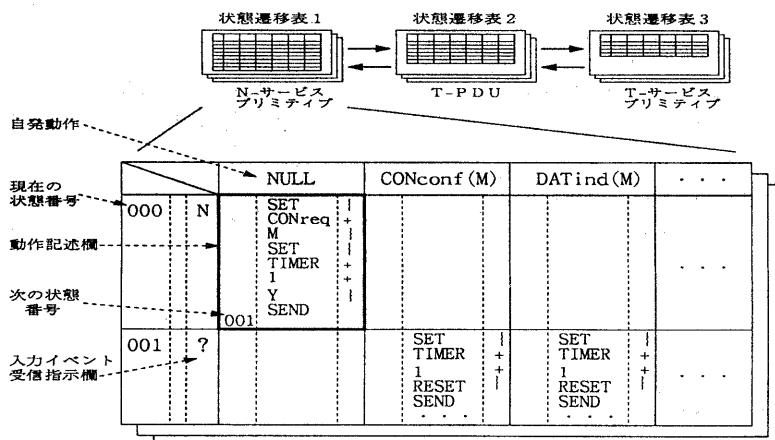


図5 ETSの構成

ント、自発動作または遷移表間遷移処理用イベントであり、縦軸は状態番号を表わし、表の各欄には、その状態で実行するアクションを記述する。なおETSは、複数の層（レイヤ）の動作に關係するので、複数の状態遷移表により構成する。すなわちイベントの発生する場所の違いによりそれぞれに状態遷移表を持ち、階層的構造を有している。この方法によりサービスプリミティブとPDU別に、試験手順、すなわち試験時の通信手順が記述できるので、着目したい点の動作がより明確に記述できる。これら状態遷移表は、下位レイヤから順に状態遷移表1、状態遷移表2、状態遷移表3、……、状態遷移表nとする。

例えば、トランスポート層では、トランスポート層エンティティ、ネットワークサービス、トランスポートサービスと複数の層の動作に關係するので、ETSは、次の3つの状態遷移表からなる。

- (1) N-サービスプリミティブの入出力用
- (2) T-PDUの入出力用
- (3) T-サービスプリミティブの入出力用

また、テストスイートインタプリタに可搬性を持たせるため、状態遷移表内では、下位機能が実行するアクションは全て記号として表わしてある。テストスイートインタプリタは、この記号が何を意味するかを知らず、状態遷移表の指示に従いその記号を下位機能に渡すだけである。

表3、表4に、動作記述欄と入力イベント受信指示欄に記述されるコマンドの種類を示す。なお、動作記述欄内の下位機能が実行するアクション（記号'CONreq'、'M'、

表3 動作記述欄用コマンドの種類

コマンド	機能
SET	行なうべきアクション命令をメモリにセット
SEND	アクション実行指示
ANSWER	上位遷移表に遷移処理結果を伝える
END	試験終了
RESULT	結果判定
DOWN	処理対象の遷移表の階層を1つ下げる
UP	処理対象の遷移表の階層を1つ上げる

表4 入力イベント受信指示欄用コマンドの種類

コマンド	機能
?	入力イベントを待つ
S	下位状態遷移表からの返答、または現入力イベントにより再度遷移表の横軸を決定
N	自発的動作

'TIMER', '1', 'Y'など）は引き数として扱われ、{}内で記述される。引き数は、|で始まり、+で連結され、|で終了する。

例えば、図5の状態遷移表1（N-サービスプリミティブ）の場合、テストスイートインタプリタは、初期状態「000」、入力イベント「NULL（自発動作を示す。）」に於いて次の動作を自発的に行なう。

(1) PMC（記号'M'）に対して、コネクション確立要求のN-サービスプリミティブ（記号'CONreq'）を発行することを下位機能に指示する（SET | CONreq + M | SEND）。

(2) 下位機能に対して、1番目の監視タイマ（記号'TIMER', '1'）を'Y'秒間動作させることを指示する（SET | TIMER + 1 + Y | SEND）。

(3) 次の状態「001」へ遷移し、'?'で示されるイベントの入力を待つ状態となる。

## 4.2 実行型試験スイートに基づく試験実行方式

ETSに基づくACTSの試験実行手順を図6に示す。

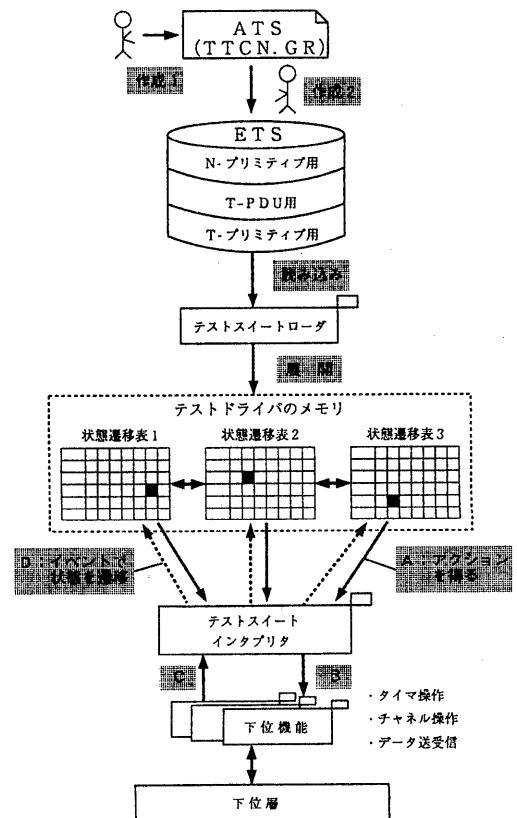


図6 試験実行手順

す。試験は、試験スイートインタプリタ方式により下記のとおり実行される。

- (1) T T C N . G R を用いて A T S を作成する。(作成1)
- (2) A T S を参照して、エディタでテキストファイル形式の E T S を作成する。(作成2)
- (3) テストスイートローダは、テキストファイル形式の E T S を読み込み、テストドライバのメモリ上に状態遷移表を階層的に分割して展開する。(「読み込み」と「展開」)
- (4) テストスイートインタプリタは、各試験ケースごとにメモリ上の状態遷移表を参照し、現在行なうべきアクションを意味する記号を得る。(矢印A)
- (5) テストスイートインタプリタは、下位機能(タイマ操作、試験チャネルおよび試験管理チャネルの操作、データ送受信)へ行なうべきアクションを意味する記号を渡す。(矢印B)
- (6) 下位機能部は、指示されたアクションを起こし、その結果をテストスイートインタプリタへイベントとして通知する。(矢印C)
- (7) テストスイートインタプリタは、下位機能からのイベントにより、階層的に分割されているそれぞれの状態遷移表の状態番号を次の状態番号に遷移させる。(矢印D)

ここで、どの状態遷移表の状態番号を遷移させるかは、階層のルートにある状態遷移表1から順に見て行き判定する。例えば、状態遷移表1内に「状態遷移表2を参照せよ(遷移表間遷移)」と記述があれば、状態遷移表1と2の状態番号を遷移する。遷移表2から3へも同様である。

- (8) 4から7までの処理を一つの試験ケースが終了するまで繰り返す。
- (9) 状態遷移表の到達した最終状態には判定が記述される。テストスイートインタプリタは、この内容からその試験ケースの試験結果として、通過、失敗、不確定あるいは試験システムの異常による試験の中止(a b o r t)のいずれかを得る。
- (10) 試験スケジュールに規定された全試験ケースに対して4から9までを行なって試験が完了する。

## 5. 試験結果

相互接続試験システム A I C T S の試験アーキテクチャに基づいて開発した適合性試験システム A C T S によりトランSPORT層クラス0の模擬I U Tとして開発

したI U Tシミュレータに対して適合性試験を行なった結果について以下に示す。なお、I U Tシミュレータは各種オプションやパラメタの値を任意に設定でき、正常あるいは異常なトランSPORT層クラス0のI U Tを模擬できるように開発したものである。

試験実行環境を図7に示す。試験システムおよび試験対象システムのプラットフォームは、ワークステーションを使用し、ワークステーション間は、通信ソフトウェア SunLink X.25 を使い X.25 プロトコルにより通信を行なうものとした。下位層の模擬通信網は公用パケット網(DDX-P)を使用した。

試験は、試験スイートの全試験ケース(40項目)を連続して行ない、その結果、全ての試験ケースに対して通過(p a s s)の判定結果が得られた。試験開始から終了までの全試験時間は28分18秒であり、試験実行時間としては妥当な値と考えられる。内訳は、試験ケース毎の試験実行時間の合計が20分55秒、試験ケース間インターバルの合計時間が7分23秒である。また、試験実行時間は、試験対象システムの動作状況を監視するための各種タイマの設定値に依存するが、設定値はすべて今回のI U Tシミュレータに対して十分な長さの10秒とした。図8に試験ケース毎の試験実行時間と実行型試験ケース(E T C)の処理ライン数を示す。試験ケースの平均実行時間は31.4秒である。E T C処理ライン数は、判定が合格となるまでに処理されたE T Cのライン数をカウンタにより計測したものであり、最小117、最大589ラインである。試験実行時間とE T C処理ライン数とは、予想通り比較的強い相関がある。

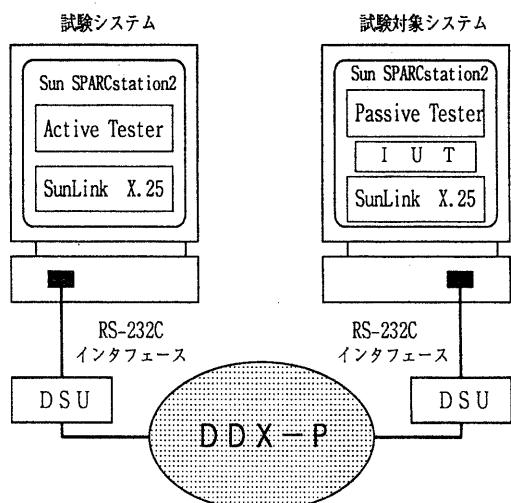


図7 試験実行環境

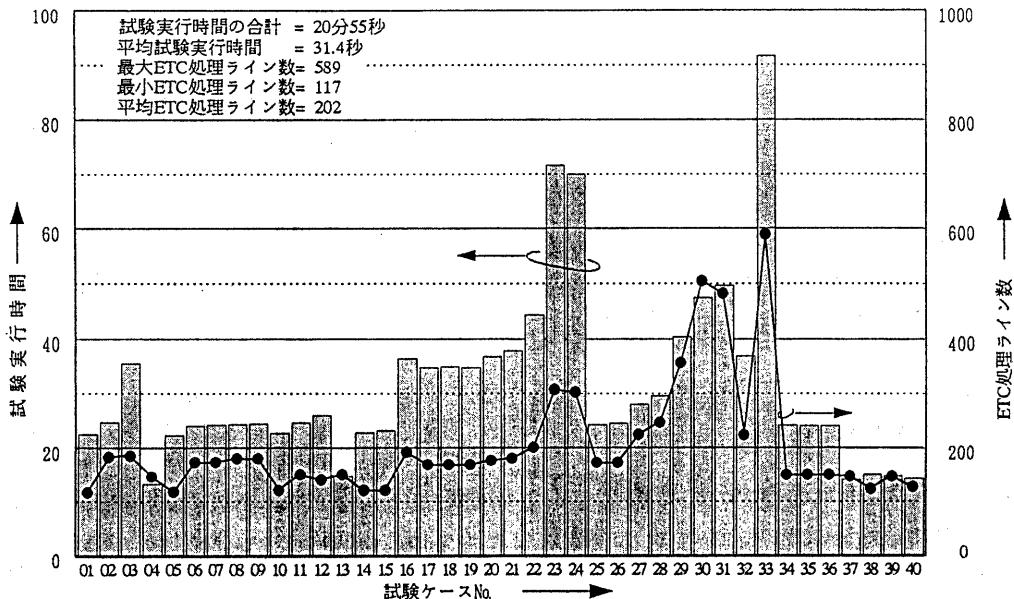


図8 試験ケース毎の試験実行時間とETC処理ライン数

## 6. おわりに

本報告では、制御観測点P C O (U) およびP C O (L) とA C T S特有の試験管理制御点P M C (M) に対する記述をT T C N . G Rによって作成した抽象試験シートについて述べ、さらに状態遷移表の構造を持つ状態遷移表形式実行型試験シートの仕様について述べた。また、試験実行方式として、状態遷移表形式実行型試験シートに従ってテストシートインタプリタが試験の実行と制御を行なう試験シートインタプリタ方式を与えた。本方式により、動的適合性試験の開始から判定までの試験実行の自動化を行なえることが確認できた。試験結果からは、試験実行時間として妥当であること、また、試験実行時間とETC処理ライン数とは、比較的強い相関関係があることなどが分かった。

今後は、相互接続試験システムA I C T S用の試験シートの開発、試験シート生成ツールの開発などを進める予定である。

なお、試験シートの網羅性および検証問題については、今後の課題と考えている。

## 謝辞

本研究に対して有益なご助言を頂いた東北大学野口正一教授、白鳥則郎教授、A I C 総方秀夫常務に深謝いたします。

## 参考文献

- [1] CCITT X.290 : "OSI Conformance Testing Methodology and Framework for CCITT Applications", (1988)
- [2] 若杉："OSI適合性検証試験の現状", 情報処理学会マルチメディア通信と分散処理研究会, 52-21 (1991.9)
- [3] G.Bochmann, R.Dssouli, J.R.Zhao : "Trace Analysis for Conformance and Arbitration Testing", IEEE Transactions on Software Engineering, Vol.15, NO.11, pp1347-1356 (1989.11)
- [4] O.Rafiq and R.Chastanet : "From conformance testing to interoperability testing", Proceedings of IFIP TC6 Third International Workshop on Protocol Test Systems (1990)
- [5] L.Lenzini, F.Zoccolini : "Interoperability tests on OSI products in the framework of the OSIRIDE-Intertest initiative", Computer Networks and ISDN Systems 24 (1992)
- [6] 中井、高橋、白鳥、野口："相互接続試験アーキテクチャの構成法", 電子情報通信学会技術研究報告, SSE90-58 (1990.7)
- [7] 後藤、似内、高橋、石幡、高橋："情報通信システム相互接続のための試験システムA I C T Sの開発", 情報処理学会マルチメディア通信と分散処理研究会 56-4, (1992.7)
- [8] ISO/IEC DIS 9464-3 : "Information technology - OSI Conformance Testing Methodology and Framework - Part3 : The Tree and Tabular Combined Notation (ITCN)", (1989)
- [9] OSTC TRANSPORT & SESSION TECHNICAL AREA TEST SPECIFICATION COMMITTEE : OSTC ABSTRACT TEST SUITE TRANSPORT CLASS 0, OSTCドキュメント (1991.1)
- [10] Tadao Wakasugi, Yoshimi Takaoka : Design and evaluation of test suites for the OSI transport layer at the conformance test center of INTAP, Proceeding of the second international symposium on interoperable information systems, pp.401-408 (1988)