

OSI試験検証システム  
— 調和試験法の実現方式の評価 —  
(INTAP試験検証技術専門委員会C法アドホック)

4H-2

天野 直己  
(株)日立製作所

若杉 忠男  
三菱電機(株)

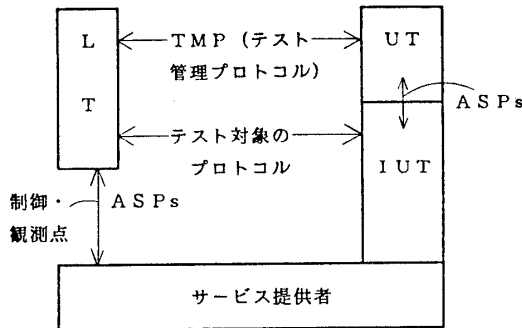
1. はじめに

INTAP (財団法人情報処理相互運用技術協会)の試験検証センターでは、OSIを実装した製品の適合性試験をするための試験システムを開発している。これまでは、遠隔試験法(R法)をベースにした試験システムの開発を行ってきたが、IUTが上位に対して行った動作をテストセンターがチェックできないなどの欠点があった。この欠点を克服するなどの理由から、INTAPでは調和試験法(C法)をベースとした試験システムの検討を開始した。  
ここではC法の実現方式について述べ、その評価結果を報告する。

2. 調和試験法(C法)

C法は、OSIプロトコルを実装した製品の試験方式で、次の特徴をもつ。(図1を参照。)

- (1) 上位テスト(UT)は、試験を受ける製品(IUT)の上に位置する。
- (2) 下位テスト(LT)は、テストセンターにある。
- (3) IUTの動作は、ネットワークを介し、テストセンター側で観測・制御を行う。LTはUT及びIUTに対するPDU(プロトコルデータ単位)を作成し、そのPDUをデータとして含む抽象サービスプリミティブ(ASP)を発行する。又LTはIUTから来るPDUを分析する。
- (4) LTとUTの同期は標準プロトコル(TMP: テスト管理プロトコル)で実現する。



[略称] LT: 下位テスト, UT: 上位テスト,  
IUT: 試験を受ける製品, ASPs: 抽象サービスプリミティブ

図1 調和試験法(C法)

C法は、他の試験方式に比べ次の利点を持つ。

- (1) R法に比べ、IUTが上位に対して行った動作までチェックできるので、詳細に試験できる。
  - (2) 被試験システム側の操作に頼らないため、試験の自動化ができる。
  - (3) リアルタイムにIUTの状態を把握でき、試験の結果を直ぐに判定できる。
- 上記の利点を持つC法をINTAPの適合性試験に適用するために、トランスポート及びセッションプロトコルのC法での試験の実現方式を検討し、その評価を行った。

3. C法の実現方式

C法を実現するには、UTとLTの同期を行なうTMPの仕様を規定する必要がある。TMPの機能としては、次のことが要請される。

- (a) LTはTMPを使用し、UTの動作を制御する。
- (b) UTはTMPを使用し、IUTの動作をLTに報告する。

ISOでは、セッション及びトランスポートプロトコルのTMPの素案(Working Draft)が提案されている。この2つのTMPは、UTとLTの同期の実現方式が異なっている。そこで、INTAPではこの2つの実現方式を検討した。次に各TMPの実現方式を述べる。

(1) トランスポートTMP

トランスポートTMPは、LTとトランスポートを実装したIUTの上のUTとの間のプロトコルである。このTMPを処理するUTは、次の3種類の内部変数を持つ(図2を参照)。

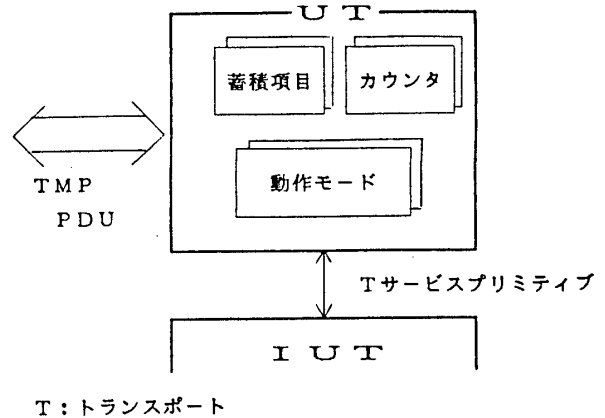


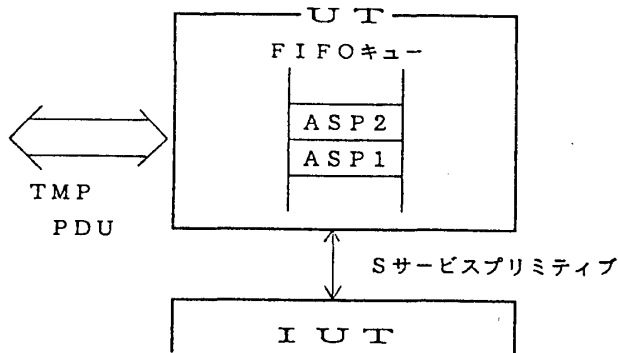
図2 C法によるトランスポートプロトコルの試験

- 動作モード--UTへの入力イベント(IUTからのASP(トランスポートサービスプリミティブ), LTからのTMPによる指示など)に対するUTの動作を決めるためのパラメタである。LTは、各イベント毎にUTの動作のリストをこのパラメタに設定する。
- カウンタ--IUTとUTの間での各ASPの発生回数や異常の回数などをカウントするためのパラメタである。
- 蓄積項目--ASPのパラメタ値を蓄積するためのパラメタである。この値は、ASP作成やLTへの報告に使用される。
- トランスポートTMP PDUは動作モードや蓄積項目の設定、各種情報の報告要求や報告といった23種類のPDUが定義されている。
- TMPを使った試験シーケンスを次に示す。
- (a) LTは利用者データにTMP PDUを含むTPDUを作成し、IUT側のシステムに転送する。
  - (b) IUTはTPDUを受信し、対応する処理を行

- ない、UTにASPを発行する。
- (c) UTはASPの利用者データにあるTMP PDUを解析し、その指示を実行する。
- (d) UTは動作モードに従いIUTへのASPの発行などの動作を行なう。もし、情報の報告が要求されているならば、このASPの利用者データに報告内容をセットしたTMP PDUを設定する。
- (e) LTはIUTから送られてきたTPDU及びその中のTMP PDUにより、IUTの動作を観測する。

(2) セッションTMP

セッションTMPは、LTとセッションプロトコルを実装したIUTの上のUTとの間のプロトコルである。このTMPを処理するUTは、コネクション毎にFIFOキューを持つ(図3を参照)。このキューの中には、セッションTMPによって運ばれたASP(セッションサービスプリミティブ)がスタックされている。



S : セッション

図3 C法によるセッションプロトコルの試験

セッションTMP PDUには、LTからUTへの指示のPDUとUTからLTへの報告の2種類のPDUから成る。TMP PDUは、ASP名及びそのパラメタ値より構成される。指示のPDUは更にそのASPを直ちに発行すべきか、キューにスタックすべきかの指定や報告のPDUの要、不要などのパラメタを含む。

セッションTMPを使った試験のシーケンス例を示す。(図4を参照。)

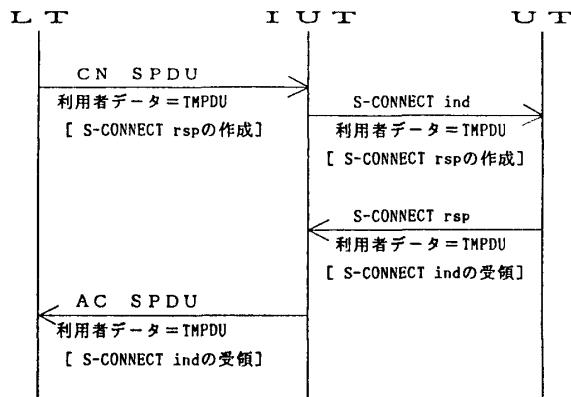


図4 C法によるセッションプロトコルのテストシーケンス例

- (a) LTは、SPDUを生成する。その利用者データには、UTが次に発行するASPを含むTMP PDUをセットする。
- (b) IUTは、SPDUを受け取り、対応する処理を行い、UTに対しASPを発行する。

- (c) UTは、TMP PDUに含まれているASPをIUTに発行する。但し、IUTからのASPがコネクションの確立やアポートなどのときのIUTの動作は別に規定されている。又、利用者データを含まないASPやTMPからの指示により、スタック内に蓄えられたASPを発行することもある。TMP PDUに報告要求が指定されていれば、UTがIUTから受けたASP及びパラメタ値を報告のTMP PDUに設定し、ASPの利用者データに入れて、IUTにASPを発行する。
- (d) LTはIUTから送られたSPDU及び利用者データ内のTMP PDUによってIUTの動作を観測する。

4. C法の実現方式の評価

2つの実現方式を項目毎に比較しその評価を行った。

(1) UTの構造

セッションのUTの構造は、内部にFIFOキューを必要とするだけであり、トランスポートTMPに比べると簡単である。

(2) TMP PDUの種類

セッションのTMPの2種類(ASPをパラメタ化している)、トランスポートの方が23種類とセッションの方が少ない。

(3) UTの動作の制御方法

セッションTMPではTMP内及びスタック内のASPにより制御する。トランスポートTMPは動作モードパラメタにより制御する。

(4) UTのASPの生成方法

セッションTMPではLTがASPを生成し、UTに送る。トランスポートTMPでは、UTが動作モードで指定されたASPを蓄積項目のパラメタ値を使って生成する。

(5) TMP PDUの転送量

セッションTMPは毎回ASPを運ぶのに対し、トランスポートTMPは、必要時のみモードの設定、蓄積項目のパラメタ値の変更を行なう。従って、トランスポートTMPの方が転送量は少ないと考える。

以上の様に2つのTMPの実現方式を比べるとテスト機能には差はないが、セッションTMPで使用している方式の方がLT、UTの内部構造が簡単となり、実装も容易であると考えられる。UTは、試験の受験者が用意する場合もあり、UTが容易に実装出来ることのメリットは大きい。従って、今後他のプロトコルに対するTMP開発にはセッションTMPの方式で開発するのが良いと考える。

5. おわりに

今後、TMPの実現方式を評価するにあたり、

- (1) TMPを使ったテストスイート(テスト項目の集まり)の作り易さを評価する必要がある。
- (2) TMPを実際に試験に使用して評価する必要がある。

これまでの検討結果を生かし、今後の試験検証システムの開発に役立てて行く予定である。

謝辞 本検討にあたりINTAP試験検証委員会委員長として御指導いただいている学術情報センターの浅野正一郎教授に感謝の意を表します。

参考文献

- [1] ISO / IEC DP9646- 1,2 Information Processing Systems - OSI Conformance Testing Methodology and Framework -Part1,Part2 (1988-01-09)
- [2] ISO / TC 97 / SC 6 / WG 4 N 336 Proposed Transport Test Management Protocol Specification (1987-12-08)
- [3] ISO / IEC JTC 1 / SC 21 / Attachment WG1 N647 Session Test Management Protocol Specification (1988 - 9月)