ＭＩＡにおけるトランザクション処理ＡＰ単体試験ツールの評価

7Ｊ－6
葛西　秀亮、水嶋　宏也
ＮＴＴ情報システム本部

1．はじめに

ＮＴＴのコンピュータ調達仕様として、ＮＴＴを中心とする７社によって規定されたＭＩＡ(Multivendor Integration Architecture)は、各ベンダの得意とする技術を有効に活用したマルチベンダシステム構築の容易化をねらいとしている。

ＭＩＡの規定の考え方は、国際標準や事実上の業界標準が存在する場合には、それらを取り入れることとしている。しかし、トランザクション処理（ＴＰ）のＡＰＩについては業界標準が存在しないため、ＭＩＡではＳＴＤＬ(Structured Transaction Definition Language：構造化トランザクション定義言語)を新たに規定した。ＭＩＡ－ＴＰのＡＰの構造を次に示す。

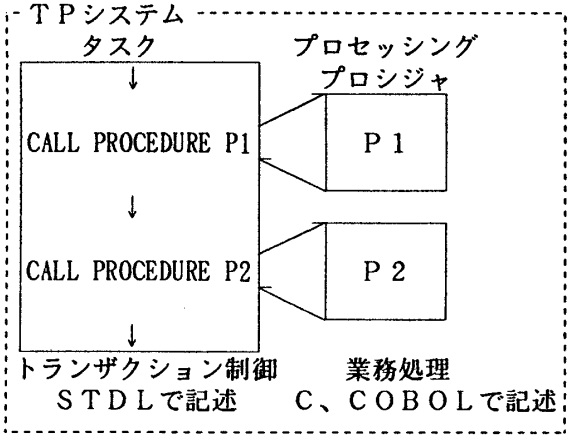

図１　ＭＩＡ－ＴＰのＡＰの構造

ＳＴＤＬの開発環境は各ベンダから提供されているが、各ベンダごとに環境や操作が異なっている。そこでＮＴＴでは、開発環境のベンダフリー化、分散化をねらいとして、Ｕｎｉｘワークステーション上でＳＴＤＬのソースプログラムの単体試験を支援するツール、ＳＴＤＬデバッガを開発した。本稿では、単体試験でＳＴＤＬデバッガを用いることの有効性について検証する。

2．ＳＴＤＬデバッガの機能と目的

ＳＴＤＬデバッガはＵｎｉｘ上のプログラムであり、ソースプログラムのシンタックスチェックを行い中間言語に翻訳するＳＴＤＬ翻訳機能、タスクのシミュレート実行を行うＳＴＤＬ実行シミュレート機能、ステップ実行やワークスペースの参照／更新等のデバッグ操作実行機能を持っている。

ＳＴＤＬデバッガの構成図を次に示す。

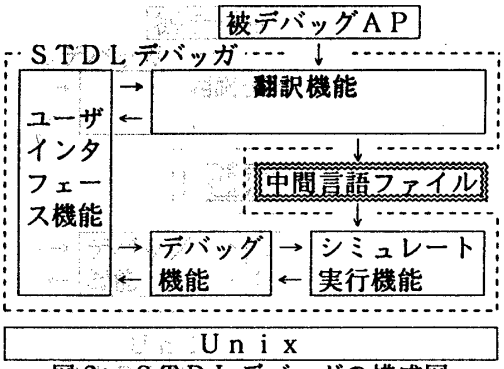

図２　ＳＴＤＬデバッガの構成図

ＳＴＤＬデバッガの導入により、ベンダ提供のデバッガと比較して次の効果が期待できる。

項目＼種別		ＳＴＤＬデバッガ	ベンダ提供デバッガ
デバッグ環境		Ｕｎｉｘワークステーション	ターゲットマシンが必要
機能充足性		制約少ない	制約多い
ユーザインタフェース	環境設定	複雑な環境設定を必要としない	環境設定に手間がかかる
	ﾕｰｻﾞのｽｷﾙ	Ｕｎｉｘ以外の専門的知識を必要としない	高いスキルが要求される

ここでは、ＳＴＤＬデバッガをベンダ提供デバッガと比較し、機能充足性、ユーザインタフェースの評価を行う。

3．機能充足性の評価

ベンダ提供のデバッガとＳＴＤＬデバッガの機能の比較を行った。比較結果を次に示す。

機能＼種別	ＳＴＤＬデバッガ	ベンダ提供デバッガ
文法チェック	○	×
ステップ実行	○	○
ブレークポイントの設定・解除	○	○
ワークスペースの参照・更新	○	○
例外発生	○	×
タスク呼出	○	○
プロシジャ呼出	×	○

An evaluation of MIA transaction AP debug tool
Hideaki Kasai　Hiroya Mizushima
NTT Information Systems Headquarters

ベンダ提供デバッガとＳＴＤＬデバッガの機能の違いが実際のデバッグ作業にどのように影響するかを定量的に評価するため、試験可能項目数の比較を行った。比較に用いた被デバッグＡＰはＵｎｉｘ－ＭＩＡの性能評価用プログラムである。プログラム規模を次に示す。

モジュール数	１３
ステップ数	７２０

このプログラムから試験項目を抽出して単体試験を実施し、試験可能項目と試験不可能項目に分類した。試験項目はプログラムロジックの１分岐あたり１項目となるよう抽出した。実施結果を次に示す。

種別／項目	ＳＴＤＬデバッガ	ベンダ提供デバッガ
試験可能項目	７７	７０
試験不可能項目	0	7

ＳＴＤＬデバッガでは全ての項目が試験可能であり、機能の充足性を検証することができたが、ベンダ提供デバッガは例外（障害事象）処理関係の７項目の試験が不可能であった。すなわち、ベンダ提供の試験環境では被デバッグＡＰを実環境と同じＴＰモニタ上で動作させるため例外を自由に発生させることができず、例外処理関係の試験項目のデバッグが困難である。この結果、例外処理ルートのバグが残留し、非摘出バグが結合試験に持ち越されるため、結合試験が困難になることが予想される。

一方、プロシジャ呼出機能と文法チェック機能は、代替手段を用いてのデバッグが可能なので問題とはならなかった。

なお、試験に用いたプログラムは、例外処理の中に分岐がなかったが、例外種別ごとに異なる処理を行うなどの例外処理に分岐を含むテストプログラムの場合、ベンダ提供デバッガでは試験不可能な項目数が更に増えることが予想される。

４．ユーザ・インタフェースの評価

ベンダ提供デバッガとＳＴＤＬデバッガによるデバッグ作業を通じて、ユーザインタフェースの比較を行った。

（１）デバッグ１サイクルの工数

ベンダ提供の試験環境でデバッグを行う場合、図３に示した複雑な手順を踏む。ＳＴＤＬデバッガを使用した場合、これらの複雑な手順を踏む必要がなくソースを変更した場合もデバッガの再起動で再デバッグが可能で、デバッグ１サイクルの工数を大幅に削減することができる。

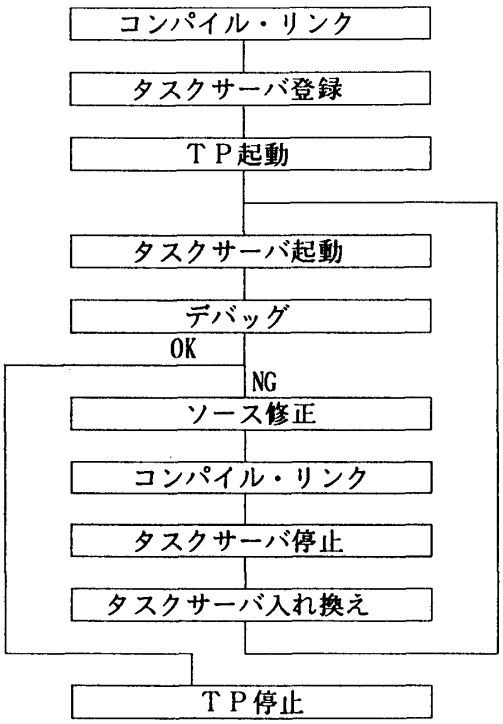

図３　ベンダ提供デバッガを用いたデバッグ手順

（２）デバッグに必要な環境設定

ベンダ提供の試験環境ではドライバやスタブの作成が必須であるが、ＳＴＤＬデバッガを用いた試験環境ではドライバの作成は必要なく、スタブを使用しない単体試験も可能である。

（３）ユーザに要求されるスキル

ベンダ提供デバッガを使用するには、ユーザにシステム管理者としてのスキルが要求される。特にＴＰシステムについての専門的知識が必要で、初心者が操作を修得するのにかなりの日数がかかる。また、ベンダごとに操作が異なり、マニュアルも初心者向けのものがないのが現状である。ＳＴＤＬデバッガを使用するには、ユーザはＵｎｉｘ以外の専門的知識を必要としない。また、操作法もマニュアルも比較的簡単で初心者も操作法を容易に修得できる。

５．まとめ

ＳＴＤＬデバッガをベンダ提供デバッガと、機能充足性とユーザインタフェースの観点で比較を行った結果、ＳＴＤＬデバッガは、機能充足性では、ベンダ提供のデバッガでは困難な例外処理ルートのデバッグが可能であるという点で優位性を検証することができた。ユーザインタフェースでは、デバッグ１サイクルに要する工数を削減できる点、操作が比較的簡易で初心者が操作法を修得しやすいという点で優位性を検証することができた。