疑似ＤＢを用いた埋込みＳＱＬを含むＡＰのデバグ方式

5G－10

芳西　崇　　中村浩司
ＮＴＴ情報通信研究所

１．はじめに

メインフレーム上で動作するアプリケーション（ＡＰ）の生産性向上のため，ワークステーション（ＷＳ）を用いた分散開発環境が整えられてきている。一般にＷＳ上ではＡＰ開発工程のコーディング，単体試験まで可能であるが，埋込みＳＱＬが記述されたデータベース（ＤＢ）アクセスを行うＡＰについてはＳＱＬが実行できないため，十分な試験が困難であった。そのためＷＳ上でＳＱＬをシミュレートするＳＱＬシミュレータ機能を提案し，効率的な単体試験方法を提案してきた。［1］

本稿では，従来の機能に，データの検索更新が可能な疑似ＤＢ機能を追加して，より実環境に近い状態での単体試験を可能とする試験方式を提案し，その実現法について述べる。

２．従来のＳＱＬシミュレータ方式

従来提案してきた埋込みＳＱＬをシミュレートする方式は，ＳＱＬプリプロセッサとＳＱＬシミュレータから構成されていた。

(1) ＳＱＬシミュレータの構成

1) ＳＱＬプリプロセッサ

埋込みＳＱＬが記述された原始テキスト・ファイル及び事前に作成されたＤＢ定義情報を入力として，埋込みＳＱＬ部をＳＱＬシミュレータが解釈できる命令に変換した原始プログラムを生成する。この際，埋込みＳＱＬのシンタックスチェックも合わせて行う。

A Debugging Method Using Dummy DB of Embedded SQL Application Programs.
Takashi Honishi and Koji Nakamura
NTT Information and Communication Systems Laboratories

2) ＳＱＬシミュレータ

ＳＱＬプリプロセッサが生成した原始プログラムとシミュレータライブラリ，シミュレータ本体から構成される。ＳＱＬシミュレータは，ＡＰが発行するＳＱＬを解析し，手入力で設定されたＳＱＬ実行結果をＡＰに返却する。実行結果設定はＳＱＬ中のホスト変数毎に可能である。

(2) ＡＰデバグ方法

①ＳＱＬプリプロセッサによりシミュレータ用のＡＰを生成する。
②ＳＱＬシミュレータ及びＡＰを起動する。
③デバッガ等によりＡＰを１行ずつ実行する。
④ＳＱＬに対しては期待する実行結果を逐次手入力しながらデバグを行う。

(3) 従来方式の問題点

従来の方式では以下に示す理由によりデバグ効率があがらないという問題点があった。

1) ＳＱＬの実行結果によりＡＰの動作が変わる場合，逐一データを設定する必要がある。
2) データの更新結果を保存できないため，ＡＰのデバグ毎に最初からＳＱＬ実行結果のデータの設定を行わなければならない。

３．疑似ＤＢを用いた方式

上記問題点を解決し，今まで実環境でしかできなかった試験をＷＳ上で実現するために従来のＳＱＬシミュレータにＤＢアクセス機能を追加することを考えた。このＤＢを疑似ＤＢと呼ぶ。

(1) 疑似ＤＢ機能の概要

1) 言語処理部

ＳＱＬの構文解析と以下の処理を行う。

①手入力処理が指定されている場合は，ＡＰに返却する実行結果をキーボード等により設定

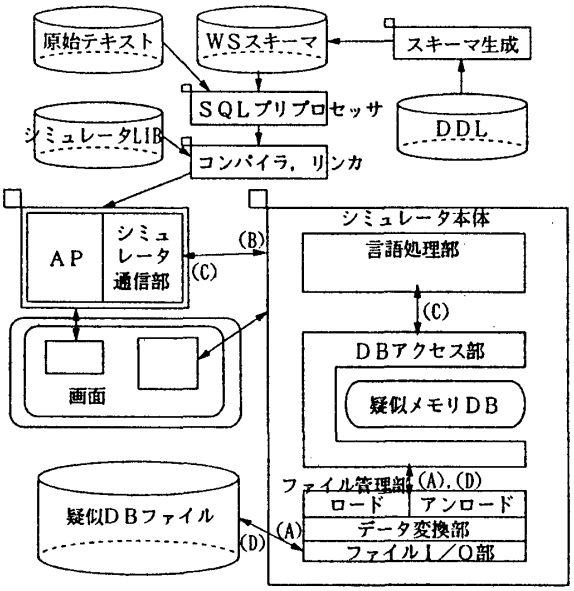

図1　SQLシミュレータ構成図

する。

②疑似ＤＢ入力処理が指定されている場合はＡＰに返却するデータを疑似メモリＤＢにアクセスして設定する。疑似メモリＤＢに対して実行できるＳＱＬは，INSERT, DELETE, UPDATE, SELECTのみであり，COMMIT, ROLLBACK 機能やリカバリは持たない。

2）疑似メモリＤＢアクセス部

メモリの確保開放及び疑似メモリＤＢの参照更新を行う。

3）疑似ＤＢファイル管理部

①アンロード：メモリ上にある疑似メモリＤＢをデータ変換機能によりテキスト形式に変換し，疑似ＤＢファイルに格納する。疑似ＤＢファイルはエディタによる修正が可能である。

②ロード：疑似ＤＢファイルからテキストデータを読み込み，疑似メモリＤＢに展開する。

(2) 疑似ＤＢを用いた場合のＳＱＬシミュレータ処理概要

1）ＳＱＬシミュレータ起動時にファイル管理部により疑似ＤＢファイル中のテキストデータを変換してメモリ上に展開する（図1(A)）。

2）ＡＰ側のシミュレータ通信部はＳＱＬをシミュレータ本体側に転送する（図1(B)）。

3）シミュレータ本体側の言語処理部は受信したＳＱＬを解析して，疑似メモリＤＢにアクセスし結果をＡＰに返却する（図1(C)）。

4）試験終了時にファイル管理部により疑似メモリＤＢを疑似ＤＢファイルに格納する（図1(D)）。

(3) 試験イメージ

疑似ＤＢを用いた場合の試験イメージを図2に示す。言語処理部では手入力処理から疑似ＤＢ入力処理への切替え機能も持っている。手入力処理から疑似ＤＢ入力処理に切替えた例を図3に示す。

4．おわりに

疑似ＤＢを用いたＳＱＬシミュレータを用いたデバグ方式について述べた。本方式を用いることにより，従来は手入力の試験のみが可能であったところ，実際のＤＢをアクセスするのと同等の試験が可能になる。これによりＷＳ上でのより多くの試験項目の消化が期待でき，現在適用を始めたところである。

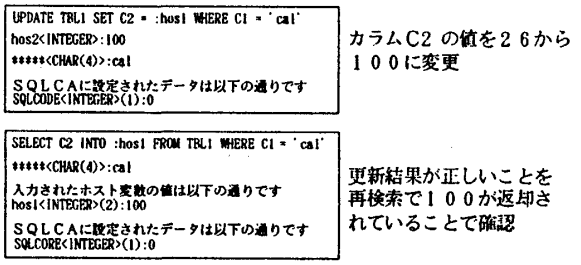

図2　疑似ＤＢを用いた試験例

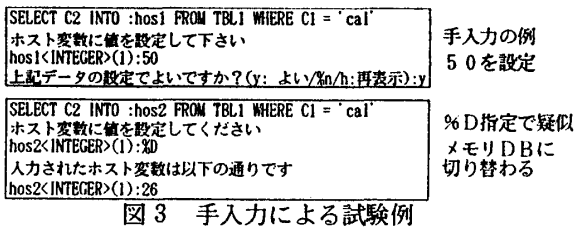

図3　手入力による試験例

<参考文献>［1］梅本佳宏，中村仁之輔：「埋込みＳＱＬが記述されたＡＰのデバグ方式」１９９０年信学会秋季全国大会