UNIXでのテスト支援環境の構築

-全体構成-

4G-3

飛山哲幸　渡邉俊雄　明智憲三郎

(三菱電機(株))

1. はじめに

UNIXオペレーティングシステムは豊富なコマンド･ツールを自由に組み合わせてソフトウェア開発を効率よく行えること、さらに他のUNIXオペレーティングシステムで開発されたツールが容易に利用できることなどから、ソフトウェア開発用として広く普及しつつある。

しかし、近年UNIXオペレーティングシステムで動作するツールとして提供されてきているものの中には、従来の専用オペレーティングシステムで動作していたツールを単純に移植してきたため、UNIXの「良さ」を殺してしまっているものも見受けられる。

当社は32ビット･スーパー･ミニコンM70MX/3000のOS60/UMX(UNIXオペレーティングシステムをもとにしたもの)上で動作する各種支援ツールを開発し充実しつつある。本論では、その中からUNIXオペレーティングシステムの「良さ」を十分に生かせる開放的なツール群より構成されているテスト支援環境について、そのコンセプトを中心に全体構成を述べる。

2. 本テスト支援システムのコンセプト

(1)開放的なシステム(UNIXとの親和性、テスト/デバッグ/カバレッジ測定の分離)

テストの準備から実行･結果確認･デバッグ･再実行･評価までの一連の作業を、UNIXの利用なしに画一的な支援を行う閉鎖的指向を排除し、UNIXの持つ豊富なコマンドを自由に組み合わせてテスト実施が出来る開放的な支援システムを構築する。「ユーザの工夫が可能であること」こそUNIX本来の姿」である。

このため、テスト/デバッグ/カバレッジ測定を分離して各ツールを単機能化し、ツールの独立性を高める。これにより、ユーザの自由なテスト環境のもとに必要なツールを選択しUNIXコマンドと組み合わせて使用できるシステムを構築する。

(2)テスト方式を制限しないシステム(テストの柔軟性)

モジュール単品テスト等にテスト方式を特定せず、プログラムの一部モジュールに欠落があっても簡単にテスト出来るシステムを構築する。また、トップダウン/ボトムアップ/サンドイッチ/ビッグバン等ユーザにマッチした何れのテスト形態にも適応し、モジュール単品テストからプログラム組み合わせテストまで連続的に支援出来るシステムを構築する。

(3)被テストモジュールのソース修正不要なシステム

トレース/スナップ等デバッグ情報を得るため等被テストモジュールのソースを修正することなしに、テスト/デバッグ/カバレッジ測定が行えるシステムを構築する。

3. 本テスト支援システムの特長

先に述べたコンセプトを基に構築したテスト支援システムの特長を以下に述べる。

- ツールが単機能に実現されており、ツール･UNIXコマンドの組み合わせ、組み替えが自由に行える。そのため、ユーザが任意なテスト環境を構成出来る。
- ドライバ/スタブはジェネレータによりターゲット言語の分かり易いソースを直接生成し、ユーザが自由に加工出来る。
- ドライバ/スタブを必要に応じて使用し、自由に複数モジュールのテストが出来る。
- テスト･モジュールを実行させるための専用環境(専用ドライバ)が不要のため、容易にプログラムテストに発展出来る。
- カバレッジ･プロファイル測定専用のツールが用意されており、テスト方式を問わず、かつ同時に複数モジュールの測定が出来る。
- 会話モードの操作･データを記憶することができ自動再実行により、再テストやカバレッジ･プロファイル測定が容易である。
- ターゲット言語(C,FORTRAN等)に依存しない共通なテスト･インターフェイスがとれる。
- ツールは全てC言語及びUNIXコマンドにて作成されており、UNIXとの親和性、移植性が高い。
- デバッガは、既存のソースレベルデバッガをそのまま使える。また、今後より高度なデバッガが供給されてきた場合に、ユーザはデバッガのみを入れ替えれば良い。これは、他のツールも同様である。

4. 本テスト支援システムの全体構成

本テスト支援システムの全体構成を図4.1に示し、その主な構成要素の説明を表4.1に述べる。

Software Test System on the Unix —Concept—
Tetsuyuki TOBIYAMA, Toshio WATANABE, Kenzaburo AKECHI
MITSUBISHI ELECTRIC CORPORATION

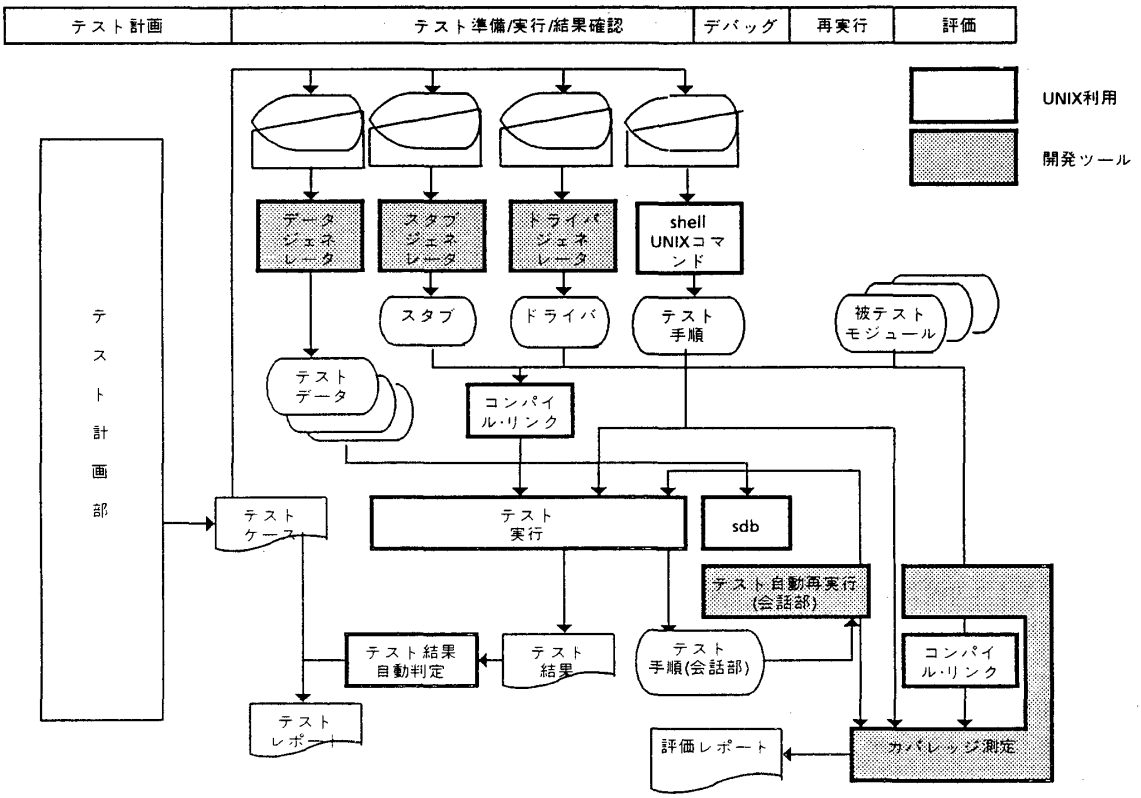

図4.1 テスト支援環境の全体構成図

表4.1 構成要素

構成要素	説 明
データジェネレータ	●構造定義文よりデータ構造を自動抽出して各種データタイプのUNIXファイルをジェネレートしテストデータを作成。
スタブジェネレータ	●NOPリターン、固定値リターン、会話モードのほかに論理式に基づく可変値リターンのスタブ・ソースをターゲット言語で会話型で生成。
ドライバジェネレータ	●会話モードのデータ設定・ダンプのドライバ・ソースをターゲット言語で会話型で生成。
shellプロシジャー及びUNIXコマンド	●UNIXコマンドを利用したshellプロシジャーを作成しテスト実行。
カバレッジ測定	●C0,C1カバレッジに加えて性能評価用の実行回数(プロファイル)も測定。測定結果の各種レポート出力。
テスト自動再実行	●テスト実行時、スタブ/ドライバの会話モードの操作を記憶し、UNIXコマンドを利用したshellプロシジャーを併用しての再実行。記憶内容の修正、あるいは初めから設定も可能。
テスト結果自動判定	●UNIXコマンドを利用し、記憶している前回の値と照合。記憶内容の修正、あるいは初めから設定も可能。
デバッガ	●デバッグはsdbまたは、専用デバッガを利用。

5. おわりに

当社においては本システムの構築以前に、UNIXの利用なしに画一的な支援を行うツールが存在していたが、あまり普及していなかった。しかし、本論にて述べたテスト支援システムは、既にC言語/FORTRAN言語のテストに広く利用されている。これは、テスト/デバッグ/カバレッジ測定を分離するというコンセプトがツール自体の使用制約を極めて小さくし、ツールを単機能で使い易くしたことが要因であるとであると思われる。

現在は、今年度末に本システムの全機能完成に向けて最終的な改良開発を行っている。また、他UNIXシステムへの移植性も十分考慮にいれた設計を行っており、ワークステーション等他機種への移植を進めていくつもりである。また、他言語への適用、テスト計画部との結合もはかっていきたい。

参考文献

渡邉ほか：UNIXでのテスト支援環境の構築ーツール概要ー、情報処理学会第33回全国大会