

テスト自動化ツールの実現

4G-7

熊谷 克夫, 萩野 宏子, 若山 正道, 北村 順次
 ㈱東芝 青柳工場

1. はじめに

一般的に、ソフトウェアの開発工数の中でテストの占める割合は40~50%と言われている^[1]。またソフトウェアのライフサイクル、つまり要求定義に始まり保守に至るまで直接“テスト”と名がつく工程は単体テスト、統合テスト、フィールドテスト等が挙げられるが、他の工程においてもウォークスルー、インスペクションといったテスト行為がなされる。

近年のワークステーション（以下ステーションと略す）の発達にともない、その強力な機能を駆使した対話型プログラムが多く作成されているが、そのようなプログラムのテストを行う場合、テスト担当者がステーションに張り付いてキー入力をしなければならないので、工数、信頼性といった面で数多くの問題が挙げられる。

本稿では、当社の分散処理コンピュータDP9080上に開発した対話型プログラムの統合テスト及び保守テストの効率化を目指したテスト自動化ツールについて報告する。

2. 自動テストの要件

統合テスト、保守テストにおける主目的は、プログラム中で変更を加えた部分だけが正しく反映されているかと、それ以外は以前とまったく同じに動作するかの確認である^[2]。その際、テスト担当者のオペレーションミス等による人的エラーを排除しなければならない。このようなテストを効果的に行うためには、次のような事柄が容易にできることが望ましい。

- (1) 過去に使用されたテストデータ及び操作手順が保存できる。
- (2) 保存されたテストデータあるいは操作手順の一部を変更できる。
- (3) 過去のテストデータ及び操作手順をもとに、再テストが自動的に実行できる。

今回実現したテスト自動化ツールは、テスト時の操作手順をテストデータとして保存することができ、それをもとに自動実行が可能となっている。

3. ツールの構成

自動テスト機能を簡単に述べると、あるテストケースに対して行ったオペレーションをすべてファイルに記録し、2度目からはオペレータを介さずにファイルからオペレーションを入力してプログラムを実行することである。これは、ステーション入出力をファイル入出力に変更することで実現されるが、この機能はUNIX*でのリダイレクション機能に相当するものである。

なお、本ツールは、自動テスト機能をもつ本体部とそこで作成されるファイルからテスト結果等を表示する支援ユーティリティで構成される(図1)。

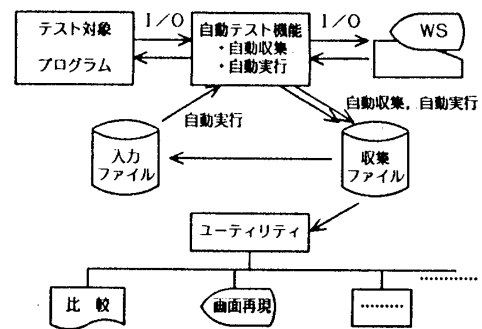


図1 テスト自動化ツールシステム構成

[1] 本体部

自動テスト機能は、自動収集、自動実行の二つの機能がある。

①自動収集機能

自動実行のテストデータを得るためテスト担当者は最初、プログラムを起動してテストを行う。このとき自動収集機能は、プログラム側がステーションに対して出す入出力要求及び入力要求に対するステーション側からの応答をすべてファイルに収集する。この際、プログラム起動時に収集ファイルを指定するだけでテスト対象は一切変更する必要はない(図2)。

Development of Automated Testing Tool

Katsuo KUMAGAI, Hiroko HAGINO, Masamichi WAKAYAMA, Junji KITAMURA
 TOSHIBA Corp.

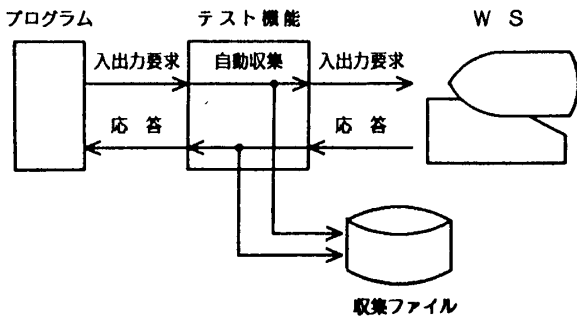


図2 自動収集

②自動実行機能

収集ファイルを入力とし、オペレータを介さずにプログラムを自動実行することができる。このとき、出力要求は通常の実行と同様にステーションに対して送出されるが、入力要求がプログラム側から出された場合ステーションに対して要求を出さずに、入力ファイルから前回の応答を読み込み、実際に要求に対してオペレーションがなされたかのようにプログラム側に応答を返す。

また、自動実行を行いながらでも、その結果を自動収集することが可能である。

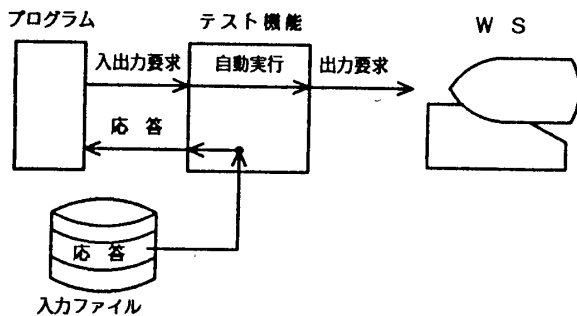


図3 自動実行

(2) 支援ユーティリティ

本体部を支援するユーティリティは、テスト実行で作成されたファイルから各種情報を出力するもので、画面再現、比較機能がある。

①画面再現機能

テスト実行時に作成されたファイルからプログラムの画面の流れを再現する機能であり、テスト対象プログラムを実行させることなく、プログラムとステーション間のI/Oをシミュレートすることができる。

本機能は実際にプログラムを動作させている訳ではないので、そのプログラムが使用するファイル等の状態をそのままにしておくことができる。また、自動実行時のモードの切り替え点の設定(自動実行から自動収集、または通常の実行)等のデータのエディットも可能である。

②比較機能

過去のテスト結果と、今回実施した結果を比較する機能である。この場合、I/Oの順番だけではなく、画面上の表示位置、入力フィールド番号等の比較を行い、どこが変更されているかを明確に表示する。

4. おわりに

今回開発したテスト自動化ツールは自動実行の局面を中心に実現したが、テスト環境として更に整備するため検証系の自動化を図る必要がある。そのほか、

- (1) ソースプログラム、オブジェクトのバージョン管理機能
- (2) モジュールテスト機能
- (3) テストカバレッジ機能

といった各種支援ツールとの統合化が必要であり、今後の課題である。

5. 参考文献

- [1] 宮本勲
“ソフトウェアエンジニアリング現状と展望”
TBS出版会
- [2] “ソフトウェア保守技術開発計画テクニカルレポートNO.1”
協同システム開発株式会社 57-10

* UNIXは、AT&Tベル研究所が開発し、AT&Tがライセンスしているオペレーティングシステムです。