

## 簡易自動回帰テストツールの試作 —RAT(Rapid Application Testing)—

古江 智和 谷口 裕一

株式会社フォーラムエイト宮崎支社 TestGroup 〒889-2155 宮崎市学園木花台西 2-1-1

E-mail: furue@forum8.co.jp taniguchi@forum8.co.jp

あらまし テスト技術者やソフトウェアの開発者自身が最小限のコストでテストを作成／運用することを主眼に置き、Windows アプリケーション用の自動回帰テストツールを試作した。自動テストツールを導入する場合テストの作成やメンテナンスに要するコスト、更にはツール自体の価格が問題になるが、今回作成したツールはこれまで筆者らが経験してきた自動テスト運用に関する反省を元に、RAT(Rapid Application Testing)を念頭に迅速、手軽な自動テストの実現を目標とした。作成に際しては、コーディング上の技術的な要件の多くを Web 上で公開されているオープンソースから取得することが出来た為、本ツールの概要と共にそれらも合わせて紹介する。

キーワード 回帰テスト, 自動テスト, RAT, オープンソース

## The Making of A Prototype for Easy Automated Regression Testing. —RAT(Rapid Application Testing)—

Tomokazu Furue Yuichi Taniguchi

† FORUM8 INC. Miyazaki Branch TestGroup 2-1-1 Gakuenkibanadai-nishi, Miyazaki, 889-2155 Japan

E-mail: † furue@forum8.co.jp taniguchi@forum8.co.jp

**Abstract** We have made a working prototype for automated regression testing. It's fundamental purpose is keeping a lid on cost for test making. This tool offers easy operation and risk-free test management from the several standpoint of cost. In a way, this tool is the antithesis of commercial testing tools. We made it with "RAT(Rapid Application Testing)" in mind. We introduces "abstract of this tool" and "coding technique and resource on WEB".

**Keyword** Regression Testing, Automated Testing, RAT, Opensource

### 1. はじめに

「マウス／キーボードのレコーダに画面キャプチャ機能を組み合わせれば簡単な回帰テストツールが作成出来る」

至極シンプルな発想をベースに本テストツールは作成された。当初は既存のフリーウェア等を組み合わせることによって実現することを試みたが、テスト技術者のみならず開発現場のプログラマでも簡単に手軽に使えることを意図した結果、Microsoft Visual C を使用して独自にツールを作成することとした。<sup>(注1)</sup>

本論では本ツールの概要を説明する他、筆者らの経験に鑑みた自動テストツールの持つ問題点にも言及する。またもう一つの目的として、同様のツールを作成する方の助けとなるべく、ツール作成に当たって参考とした Web 上の幾つかのオープンソースを紹介する。

更にソフトウェアテストに役立つその他のフリーウェア／シェアウェアも何本か紹介する。尚、本ツールは'TestFinder'の名称でフリーウェアとして社外にも提供中である。<sup>[1]</sup>

### 1.1. 自動テストツールの問題点

弊社ではこの三年間、商用の自動テストツールを用いて開発製品のテストを行ってきた。その間、自動テストツールの運用コスト（作成／メンテナンス）を下げるべく、また開発現場（開発者自身）でもツールを運用出来るような方策を模索してきたが、下記の様な問題点を解消するに到っていない。

- ・ テストの作成／メンテナンスに費やされるコストの低減が困難
- ・ ツールの習熟に時間を要するため、開発者が使用するには敷居が高い  
(結局熟練したテスト技術者が必要)
- ・ テストツール自体のコストパフォーマンスの問題  
(高価過ぎる)

上記問題点に対する解決策の一つとして今回、GUI 上で得られる結果に的を絞った自動回帰テストツールを作成した。

### 1.2. RAT Tool としての要件

'RAT(Rapid Application Testing)'はさして馴染みのな

い用語であるが、本論では今回作成したツールの特徴を示すために'誰でも直ぐに簡単にできるテスト'ぐらいの意味合いで使っている。(注2)

今回のツールの性格としてはいわゆる'RAT Tool'であることを第一とし、テストの作成/実行に対して

- ・特別な知識を必要としないこと (検証も含めて運用が簡単であること)
- ・最小限のコストで作成できること
- ・作成したテストがメンテナンスフリーであることを満たすべき要件とした。

(注1) UWSC(5章参照)からキャプチャソフトを呼び出すようにすれば簡単に実現できそうである。

(注2) もちろん'RAD(Rapid Application Development)'から派生した語句であることは容易に想像できる。

## 2. ツールの概要

本ツールは下記に示す各フェーズを経て回帰テストを実現する。

- ①テスト作成 - ユーザが Windows 上で行う操作を自動記録. 画面キャプチャ操作を含む.
- ②テスト実行 - ジャーナルを読んで, テスト作成時の動作を正確に自動再生. 記録時と同タイミングで画面キャプチャ自動実行
- ③テスト検証 - 作成/実行時にコレクションされたキャプチャファイルを比較し, テスト結果として提供.

図1にテストフローを示し, 以降の節で各フェーズの特長を説明する。

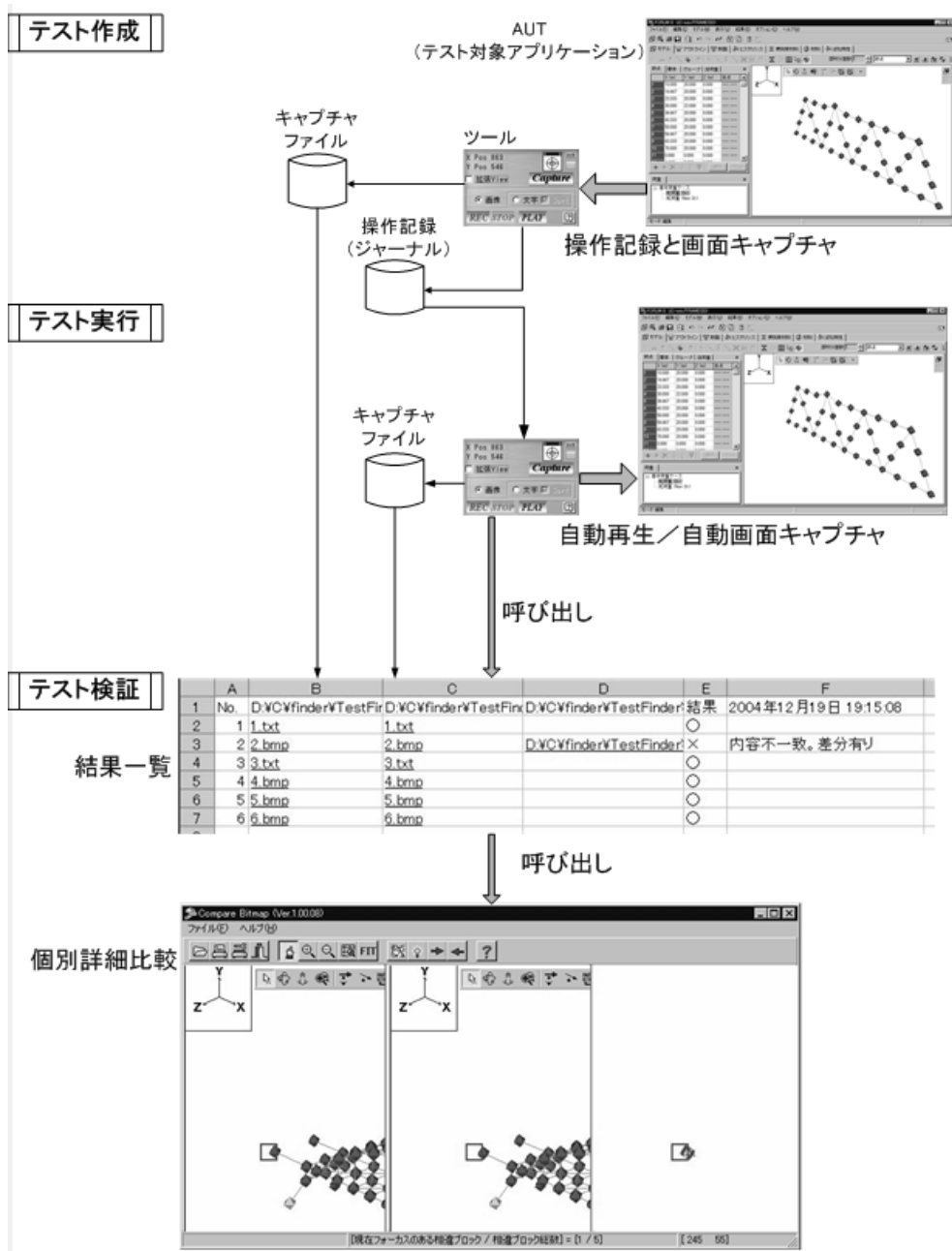


図1 テストフロー

## 2.1. テストの作成と実行

本ツールの概観を図2に示す。

REC ボタンを押下すると記録を開始するので、その間ユーザは AUT を操作しながら任意のタイミングで任意のウィンドウを Capture ボタンでキャプチャする。

STOP ボタンを押下すると操作記録(以降ジャーナルと記述する)が保存され、PLAY ボタンを押下するとジャーナルを Capture 処理も含めて自動再生する。

操作の記録再生方法としてはアナログ方式を採用している。すなわちマウスやキーボードの動作状態を座標値や時系列も含めて完全にエミュレートすることを指しており、ドラッグ&ドロップ等の操作も含め全ての操作を正確に再現する。反面、不要な動作も再現するため自動再生動作がやや冗長になり易い。

また、スクリプト等の提供によるジャーナルのメンテナンスはまったく意図していない。これはスクリプトのメンテナンスが自動テストツールの運用上、極めて大きなコストになることが筆者らの経験上分かっているため、本ツールのアーキテクチャからは完全に切り捨てた。従って AUT(Application Under Test)の GUI 上の変更が発生するとそのジャーナルは使用できなくなる可能性が高い。それでも一から作り直した方が低コストで済む事の方が多い。

テスト作成時にはおおむね以下のような点に留意する。

- Application の起動時から記録しない。起動後のメイン画面の状態から記録開始する。
- 可能であればどの Window も全画面状態にして記録する。画面位置、サイズの変更を防ぐ。
- 時間を要する処理の後には一定のタイムラグを意図的に取った後操作再開する。記録/再生時の環境の違いにより処理時間も異なる可能性がある。
- タスクバーは使用しない。記録/再生時に起動中の他 Application は各々異なっている可能性がある。

## 2.2. テストの検証

検証には別途作成した補助ツールを使用する。

テストの作成/実行時に各々キャプチャされたファイル群を比較し、差分の有無をレポートする。検証機能は ExcelVBA の AddIn 形式(.xla)で作成し、ツール本体から呼び出すことで実現している。(図3 参照)

また弊社の製品は精細な図面を出力するケースもあるため、独自の比較機能を有する画像比較ツールも別途提供している。(図4 参照)

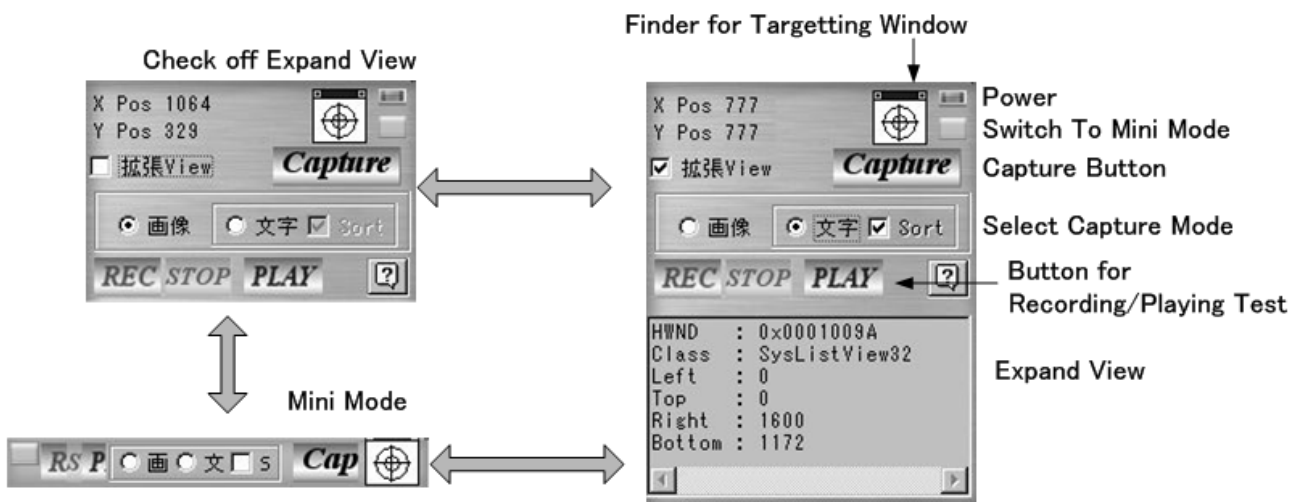


図2 ツール本体概観

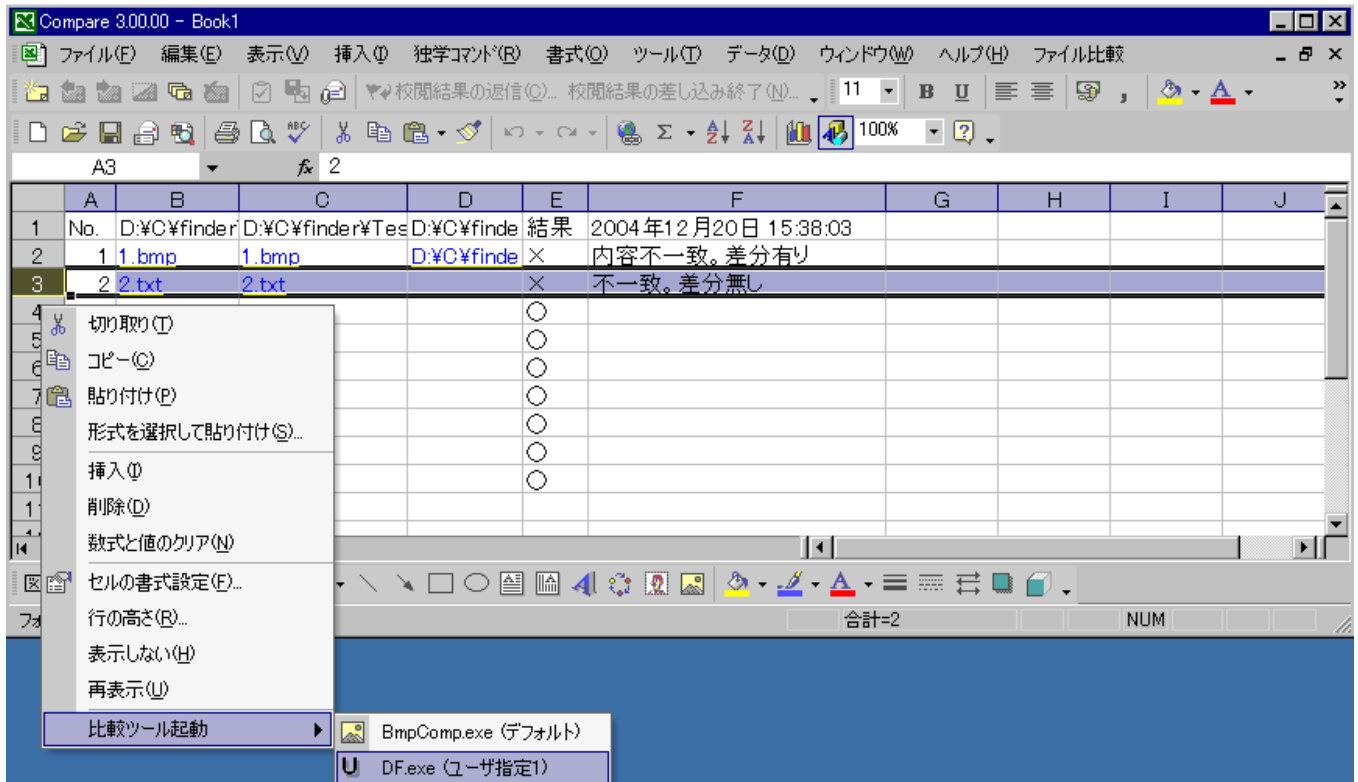
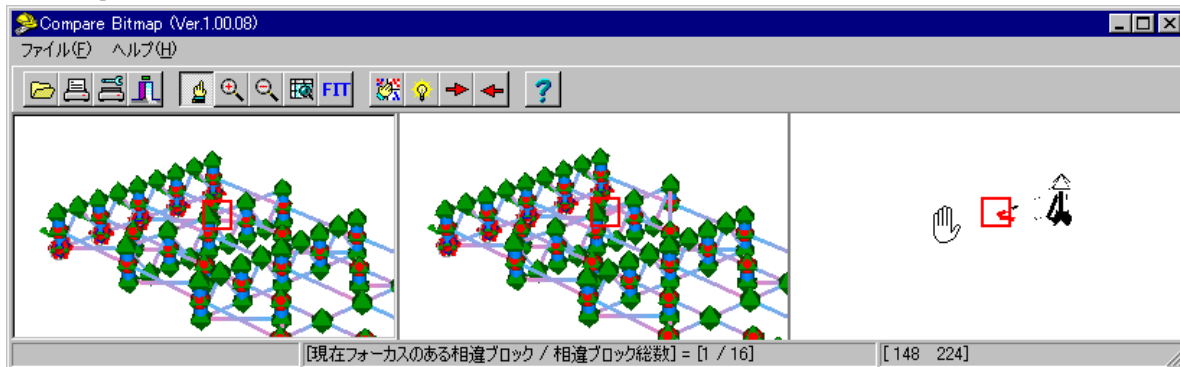


図3 テスト結果一覧

Bitmap 比較ツールの呼び出し



Text 比較ツール(DF.exe)<sup>[2]</sup> の呼び出し

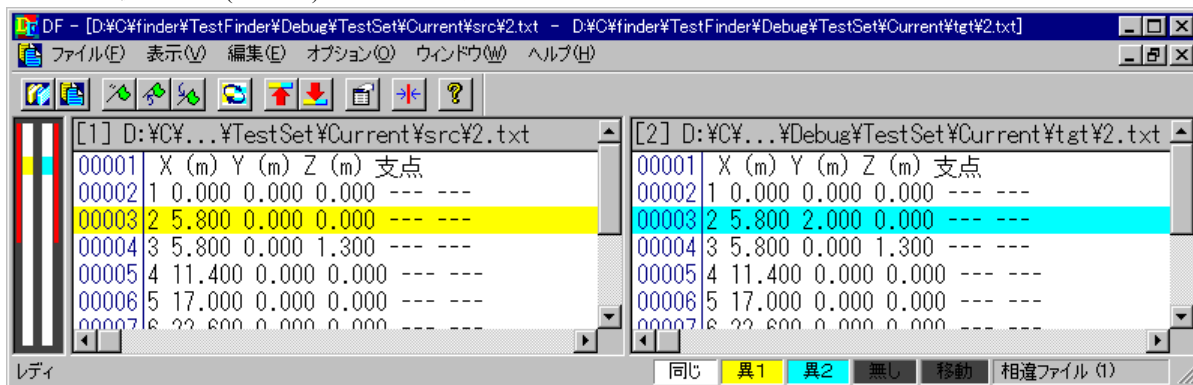


図4 個別詳細比較

### 3. 技術的ハイライト

類似のツールを独自に作成されたい方のために、ここでは本ツール作成にあたって利用した WEB 上のオープンソース等を紹介する。尚、開発言語は VisualC の SDK をベースとしている。

#### 3.1. 操作記録再生

マウスとキーボード操作の記録／再生は、Win32API 自体に機能が用意されており実装は非常に容易である。具体的には **SetWindowsHookEx** 関数の第一パラメータに **WH\_JOURNALRECORD** か **WH\_JOURNALPLAYBACK** を指定することにより実現している。お手本としては "猫でもわかるプログラミング Windows SDK 編"<sup>[3]</sup>の第 164,165 章あたりが理解し易い。また、この関数の実装に際しては下記の点に留意されたい。

- ① グローバルフック関数であるため、DLL として実装する必要がある。
- ② マウスホイールの動作は記録再生できない。Message 自体は **WM\_MOUSEWHEEL** として取得出来るがホイールさせた方向及び移動量を正しく取得することが出来ない。記録自体は別途 **WH\_MOUSE\_LL** をフックすれば可能と思われるが、**JournalPlaybackProc** を使用している限りにおいては再生動作の実現が不可能である。本ツール上でホイール動作を記録した場合の再生動作は全く動作しないか常に下方向にスクロールするか、のどちらかとなる。

#### 3.2. キャプチャ

Window を Bitmap として取得する手法は容易である。TargetWindow の DC を **CreateCompatibleBitmap** に渡してやれば良い。

対して Target の文字列取得には工夫が必要になってくる。手法としては文字の形状をパターン認識して判別する OCR 的な方法等も考えられるが、本ツールでは DLL Injection と呼ばれる方法を使用した。動作としては例えば、Target が使用している GDI32.dll の **ExtTextOutW** を自前の関数に hook しておき、TargetWindow に再描画命令 (**RedrawWindow**) を発行すると **ExtTextOutW** のパラメータとして描画文字列を受け取ることが出来る。筆者らはこの hook を安全に行うメカニズムに対する理解が未だ不十分であるが、フリーウェア DokoPop! の作者が公開されているオープンソースのプログラム<sup>[4]</sup>を真似て何とか実装することが出来たのでそちらを参照されたい。

また、"Advanced Windows"<sup>[5]</sup>の第 22 章「DLL の注入と API フック」に詳述されている。

#### 3.3. ユーザインタフェース

本ツールの GUI はダイアログベースで提供しており格別困難な部分はないが、ターゲットを特定するための SPY++ 風のインタフェースについては、Web の CodeProject にあった Article<sup>[6]</sup>を元に作成した。

また MFC をベースに作成したい場合は、同じサイトの別の Article<sup>[7]</sup>を参考にして欲しい。

#### 3.4. ジャーナル

本ツールで作成される "テスト" (またはジャーナル) と称しているものは、3.1 でも述べた通り **SetWindowHookEx** 関数のフックタイプとして **WH\_JOURNALRECORD** パラメータを指定した後、**JournalRecordProc** コールバック関数が受け取る 3 番目の引数 (LPARAM) を **EVENTMSG** 構造体でキャストして得られたものをそのまま CSV 形式で保存したピュアなテキストファイルに過ぎない。(図 5 参照)

CSV 形式にした理由は当初、Excel 等の SpreadSheet に読むことによりジャーナルを編集する意図があったためである。但し、幾つかのルールを作成してプログラムでジャーナルを最適化 (実行時間の短縮、不要なマウス動作のカット等) すること自体は容易であるが、テストの再生に失敗する可能性が大きくなるため、現時点ではジャーナルに手は加えていない。(注 1)

hwnd	message	paramH	paramL	Time	
	A	B	C	D	E
52	0	512	1020	1451	6184046
53	0	512	1022	1452	6184062
54	0	512	1025	1453	6184078
55	0	512	1027	1453	6184093
56	0	512	1028	1453	6184281
57	721336	513	1028	1453	6184281
58	0	512	1028	1452	6184453
59	0	512	1029	1450	6184484
60	0	512	1029	1448	6184515
61	0	512	1029	1448	6184515
62	0	512	1027	1443	6184531
63	0	512	1024	1427	6184546
64	0	512	1021	1409	6184562
65	0	512	1017	1385	6184578

図 5 ジャーナルファイル

(注 1) 例えば OS のスタートメニューからマウスを用いてサブメニューを辿ってき、あるアイテムを選択するような動作を再生する場合、この間の **WM\_MOUSEMOVE** メッセージはほとんど全部必要であるが、別の不要な切捨てるべき **WM\_MOUSEMOVE** との区別は簡単ではない。

## 4. ツールの活用

弊社の TestGroup 内では本ツールを画面出力を対象とした回帰テストに使用しているが、特に以下のようなケースにおいて効果的である。

- ・ **新たにテスト対象のラインアップに加わったプログラム**

通常、作成コストの関係から次回リリース時までにテストを作成して置き、次版出荷時に初めてテストを実施するようなケースであるが、必要であれば当日中にでもテストの作成／実行までを実施できる。

- ・ **将来においてユーザインタフェースの大改訂が想定されるプログラム**

通常、正規のリリース版が出るまでは自動テストの導入は見合わせるが、α版の段階においても必要であれば、ビルド間の相違をチェックするためだけにテストを作成することもできる。

これらは何れも、本ツールを使用した場合のテスト作成コストが極めて小さい(時間がかからない、知識を必要としない)ことから来る利点である。本ツールで作成されるテストは場合によっては使い捨てでも構わない。すなわち一回限りの検証の為に作成しても、コスト面で不利にならない手軽さを有している。また、AUTのユーザインタフェースにさえ変更が無ければ長期間使用することも可能である。

## 5. その他のツール紹介

最後にテスト構築に役立つようなフリー／シェアウェアのツールやライブラリを幾つか紹介する。

- ・ UWSC (Freeware) <sup>[8]</sup>

概要：Windows 操作の自動化、マウスの自動移動。  
特徴：強力なスクリプトエンジンを備えており、幅広いカスタマイズが可能

- ・ SnagIt (Shareware) <sup>[9]</sup>

概要：Image, Text, Video, Printer 等多くの形式でのキャプチャを行えるソフト。  
特徴：Video Capture は不具合等の手順再現に対して高い有用性を持つ。TextCapture 機能有

- ・ NNScanText (Freeware) <sup>[10]</sup>

概要：文字列専門のキャプチャソフト。  
特徴：Text Capture に関しては最も優れた機能を有しているソフトの一つ

- ・ madCodeHook (商用外利用 Free) <sup>[11]</sup>

概要：Sytemwide/API Hooking と DLL Injection を実現するためのライブラリ。

特徴：Text Capture 機能の実装等に利用可能

## 6. 提言

冒頭にも述べたとおり、本ツールの機能は極めてシンプルであり、また作成も容易である。恐らく既存のフリーウェアやシェアウェアを組み合わせても実現出来る可能性の高い、ごく単純なツールと言えよう。

考慮すべきは何年間か高価な自動テストツールを運用してきた筆者らが、その経験の末にこのようなシンプルなツールに対して、商用自動テストツールと似たような使用価値を見出し、かつ Test に使用しているという点にある。

テストの自動化を検討する企業やユーザは、自動化導入によるコスト(価格とテスト作成／実施)が本当に見合うものであるか、自らの手で十分な時間をかけて吟味されることをお奨めする。

## 文 献

- [1] <http://www.vector.co.jp/soft/winnt/prog/se347531.html>
- [2] <http://www.vector.co.jp/soft/win95/util/se113286.html>
- [3] [http://www.kumei.ne.jp/c\\_lang/index\\_sdk2.html](http://www.kumei.ne.jp/c_lang/index_sdk2.html)
- [4] "DCHook sample program".  
<http://homepage3.nifty.com/TaN/prog-tips.html>
- [5] Jeffrey Richter, Advanced Windows 改訂第4版, 長尾高弘(翻), 株式会社アスキー, 東京 2004
- [6] "MS Spy++ style Window Finder"  
<http://www.codeproject.com/dialog/windowfinder.asp>
- [7] "Screen capture, window resizing utility with source code"  
[http://www.codeproject.com/tools/WindowScraping\\_Utility.asp](http://www.codeproject.com/tools/WindowScraping_Utility.asp)
- [8] <http://www.vector.co.jp/soft/win95/util/se115105.html>
- [9] <http://www.techsmith.com/>
- [10] <http://homepage2.nifty.com/fukehara/index.htm>
- [11] <http://www.madshi.net/>