Framework 及び Excel を使用した自動テスト効率化試行

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 自動テストツール WinRunner に,EMOS Framework 及び Microsoft Excel を組み合わせて作成,実施したテスト事例を紹介する。EMOS Framework を組み込むことにより「自動テストメンテナンスの効率化」を、Excel VBA との連係により「柔軟なテスト結果比較」を達成することを目的とした。

キーワード 自動テスト, テストメンテナンス, テスト結果, WinRunner, EMOS Framework, Excel

Approach for efficient Automated Testing with Framework and Excel

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 It introduces the test case of WinRunner with EMOS Framework and Microsoft Excel. This test case has two purposes. One of them is 'Efficent maintenance for Automated Testing'. We have incorporated EMOS Framework in Winrunner. And another is 'Flexible comparing of test results'. We have use of Microsoft Excel for it.

Keyword Atutomated Testing. Test Maintenance, Test results, WinRunner, EMOS Framework, Excel

1. まえがき

弊社ではおよそ２年前から自動テストツールを使用して開発製品のテストを行っているが，自動テストの作成及び実施を繰り返す中で下記に示すような問題が次第に顕著になっていった．

・製品更新時の自動テストメンテナンス工数
・自動テスト結果（特に数値比較結果）の柔軟性の欠如

本論文では上記問題の改善を目的として作成した自動テスト事例を紹介する．

2.ではまず自動テストツールに組み込んだ EMOS Framework の概要を説明する．3.では本テスト事例の概要を TestCycle に沿って説明する．4.ではテスト結果比較に用いた手法について説明し，5.ではテストメンテナンス工数削減のために使用した手法を解説する．

尚，本事例で使用したツール類及び AUT(Application Under Test)は下記の通り．

・Marcury Intaractive WinRunner 7.50
・Microsoft Excel 2000
・EMOS Framework Winrunner Add In 1.4.0
・Forum8 UC-win/Section Ver1.0.0

2. EMOS Framework の概要

EMOS Framework は Free の WinRunner Add-in でインターネット上から誰でも取得することが出来る(1)．インストール後は WinRunner 起動時に追加アドインとして組み込むことで利用可能になり，以下のような特徴を持つ．

・テストデータ及びテストフローの制御を TestTable(Excel)から取得してテストを実行する
・Framework の仕様を意識してテストスクリプトをデザインすることにより，メンテナンス効率の高い自動テストセットを作成することができる．

EMOS Framework の System 構造(2)を図 1 に示す．

ユーザ(テスト作成者)は EMOS Framework が提供する Template を参考にスクリプトを構築し，テストフロー及びテストで使用する要素(キャプション，ボタン名称等)を Excel WorkSheet にテーブルとして作成する．テーブル例を図 2 に示す．

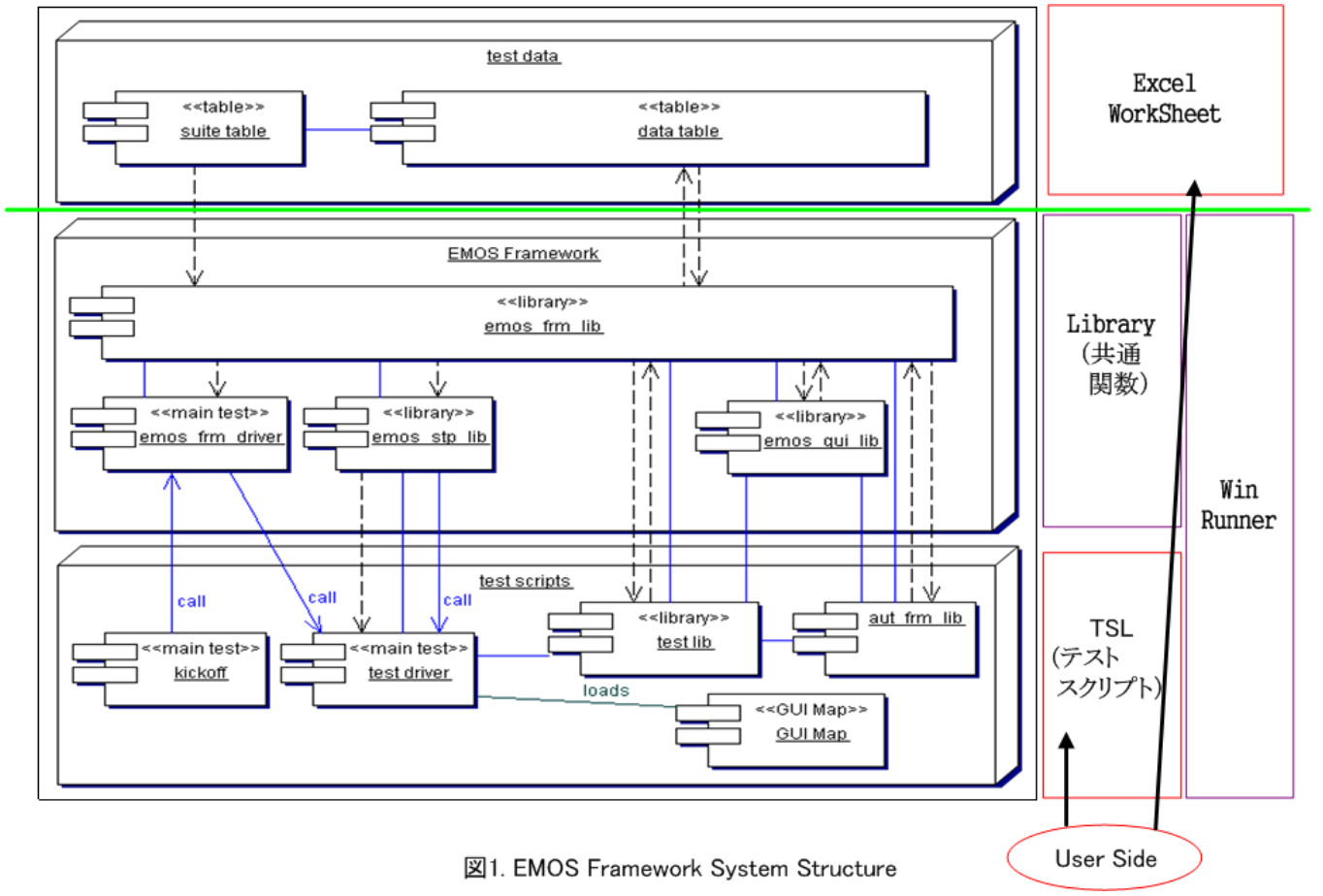


図1. EMOS Framework System Structure

Microsoft Excel - T4_SectionM1.xls [共有]

G38

	A	B	C	D	E	F	G
1	IDX	Name	Description	1	Con1	Con2	Con3
2	x	Description	this row defines test case descriptions	Make Test Section 1	BASE断面(Con1)	BASE断面(Con2)	BASE断面(Con3)
3	x	Testsequence	this row defines the sequence for the execution of test blocks in a test case	LINK""Con1 LINK""Con2 LINK""Con3 LINK""101 LINK""102 LINK""103 LINK""201 LINK""202 LINK""302 LINK""304	MainTab MainSpdBtn CreateNew LINK""T4_SectionD1.xls "Con1	MainTab MainSpdBtn CreateNew LINK""T4_SectionD1.xls "Con2	MainTab MainSpdBtn CreateNew LINK""T4_SectionD1.xls "Con3
4	x	MainTab	Tab選択				
5		label	Tab文字列		Section	Section	Section
6		expval	期待値 ○|× (check mode)		<<Clear>>	<<Clear>>	<<Clear>>
7	x	Popup	メニュー選択				
8		menu1	menu文字列				
9		menu2	sub menu文字列				
10		menu3	sub sub menu文字列				
11		expr	メニューを有効にする為のStatement				
12		expval	期待値 ○|△|× (check mode)				
13	x	MainSpdBtn	処理選択(CHK=チェックモード)				
14		tbar	ツールバー分類名		Section	Section	Section
15		btn	ボタン名(ボタン機能名)		CREATE	CREATE	CREATE
16		expval	期待値 ○|△		<<Clear>>	<<Clear>>	<<Clear>>
17	x	CreateNew	新規名称				
18		label	ダイアログキャプション		New Section	New Section	New Section
19		name	名称		Con1	Con2	Con3
20		ddlb	comboboxエントリ		SFHB-Ⅲ	SFHB-Ⅲ	SFHB-Ⅲ
21		button	押下するボタン名		OK	OK	OK
22							

←Test ID

Test Sequence
テストのフロー
を記述

変
数

Function
Arg 1
Arg 2
Arg n

図2. TestTable

3. テスト事例の概要

AUT(Application Under Test)には 1.で述べた通り弊社の製品 UC-win/Section(Ver1.00.00)を使用した．テスト内容は，AUT 内で種々の材料や形状毎に断面を作成，計算後に得られる部材力等が結果値として妥当かを検査するものである．

本テスト事例の Test Cycle を図 3 に示す．

本テスト事例では Excel WorkSheet を異なる 2 つの目的に使用していることに留意されたい．

(1)Test 結果比較用(期待値 WorkSheet)

結果を取得したいテストの要素名(ここでは断面 ID)と,結果期待値からなるリストとして構成されている．

シート例を図 4 に示す．

本テスト事例では,ここに記述されている要素のテスト結果のみを取得し，同じシートにテスト結果値を取り込んでいる．これらの WorkSheet の I/O には Framework は使用していない．また各 WorkSheet ファイルには結果比較に使用するための VBA コード(マクロ)が含まれている．

(2)Test 制御

図 3 の②(右側)に示される WorkSheet は Framework の TestTable として供されている．目的は主にテストフローの制御，GUI 周りの変数(ボタン,メニュー名称等)及び入力パラメータの提供である．内容は図 2 を参照されたい．

4. テスト結果の比較

結果比較において下記の要求事項を満たすような機能を ExcelVBA で作成した．

(1)許容誤差の設定

ある程度の誤差は結果 OK と見なすよう，許容誤差をユーザが設定可能とした．

(2)比較結果の Visual 化

比較結果において，期待値との完全一致,許容誤差内一致及び不一致の 3 種類が視覚的に判別出来る様な機能を設定した．

テスト結果例を図 5 に示す．

これらの機能は結果シートにマクロとして含まれており，結果を検討するユーザはこのシートを直接操作して検討することが出来る．

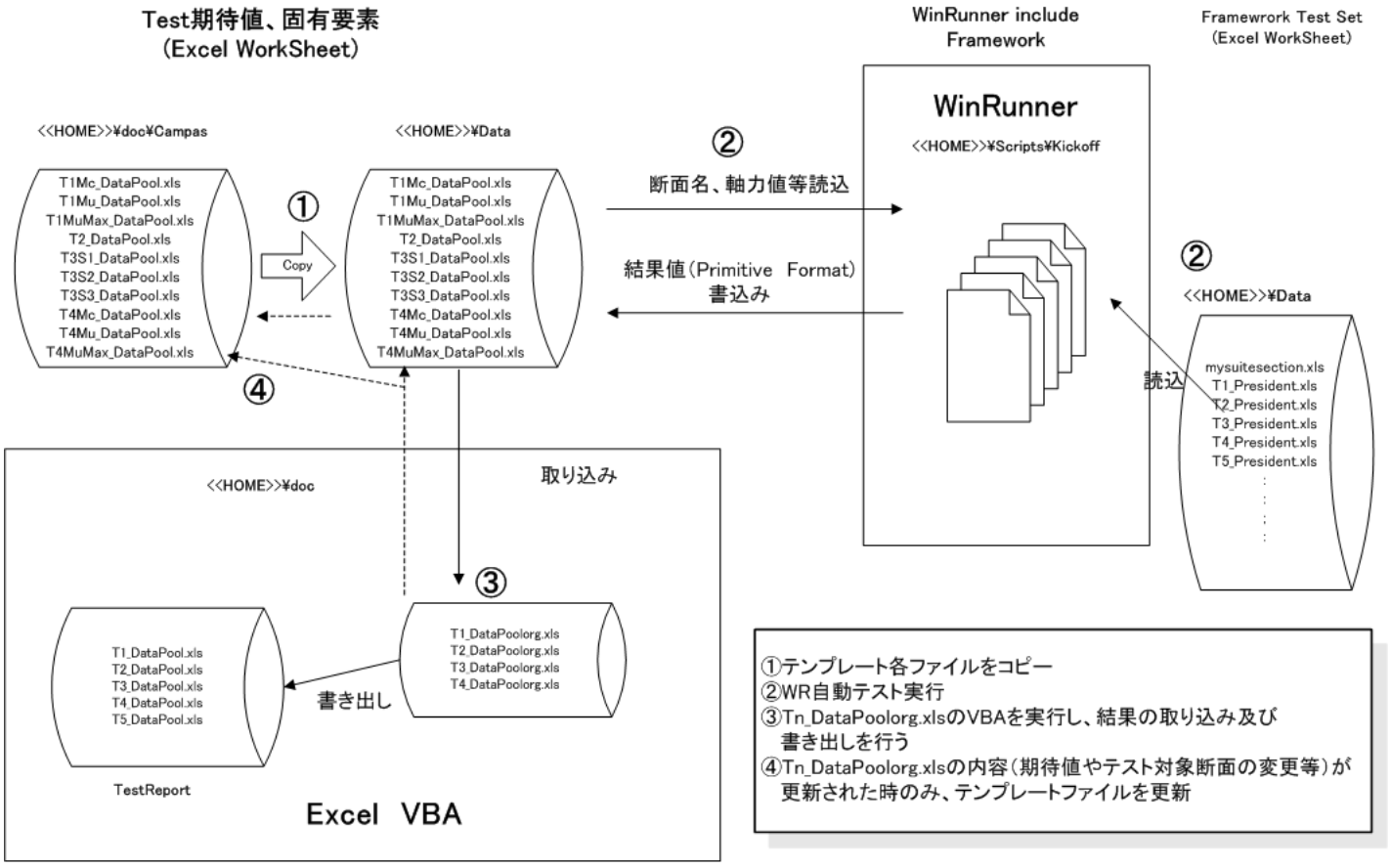


図3. Test Cycle

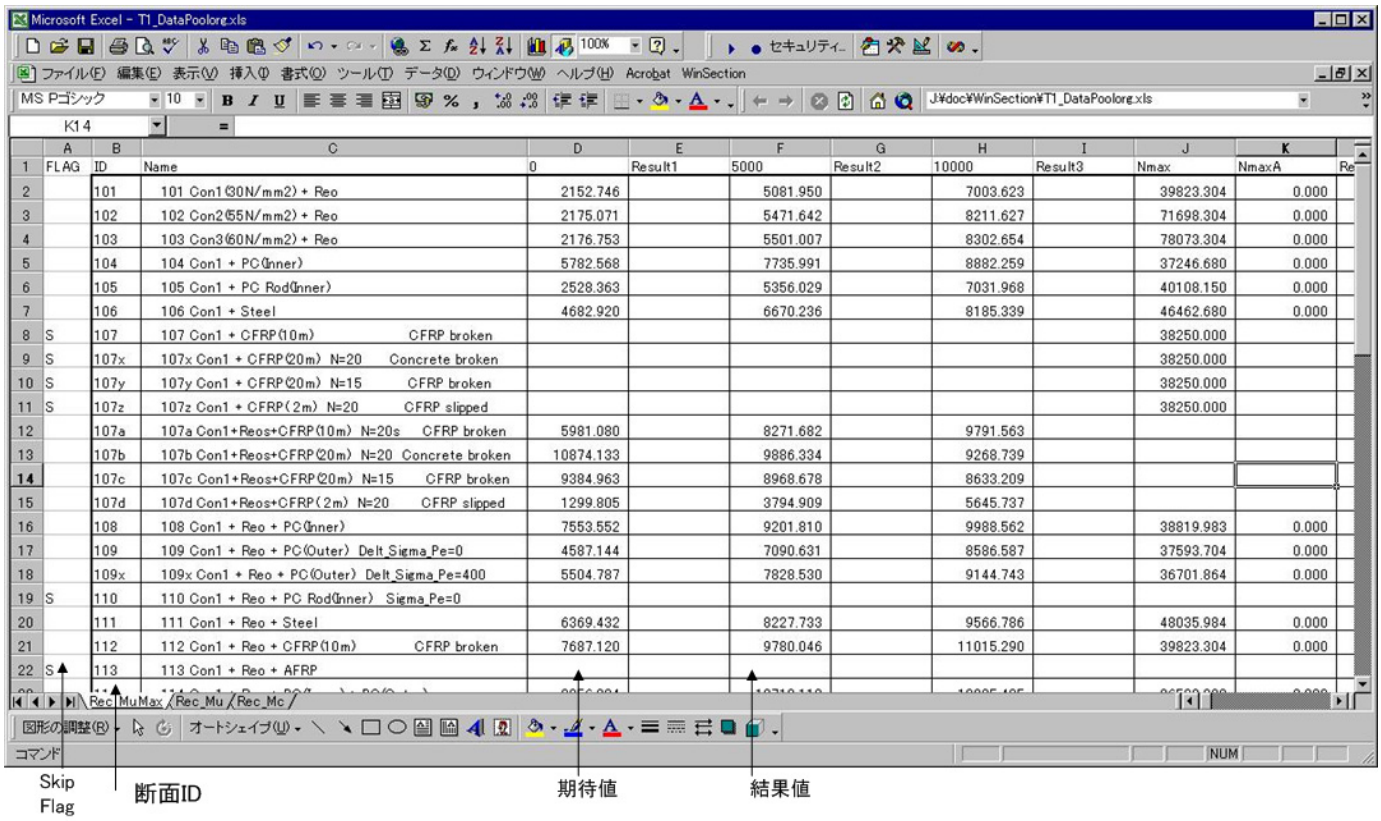


図4. 期待値WorkSheet

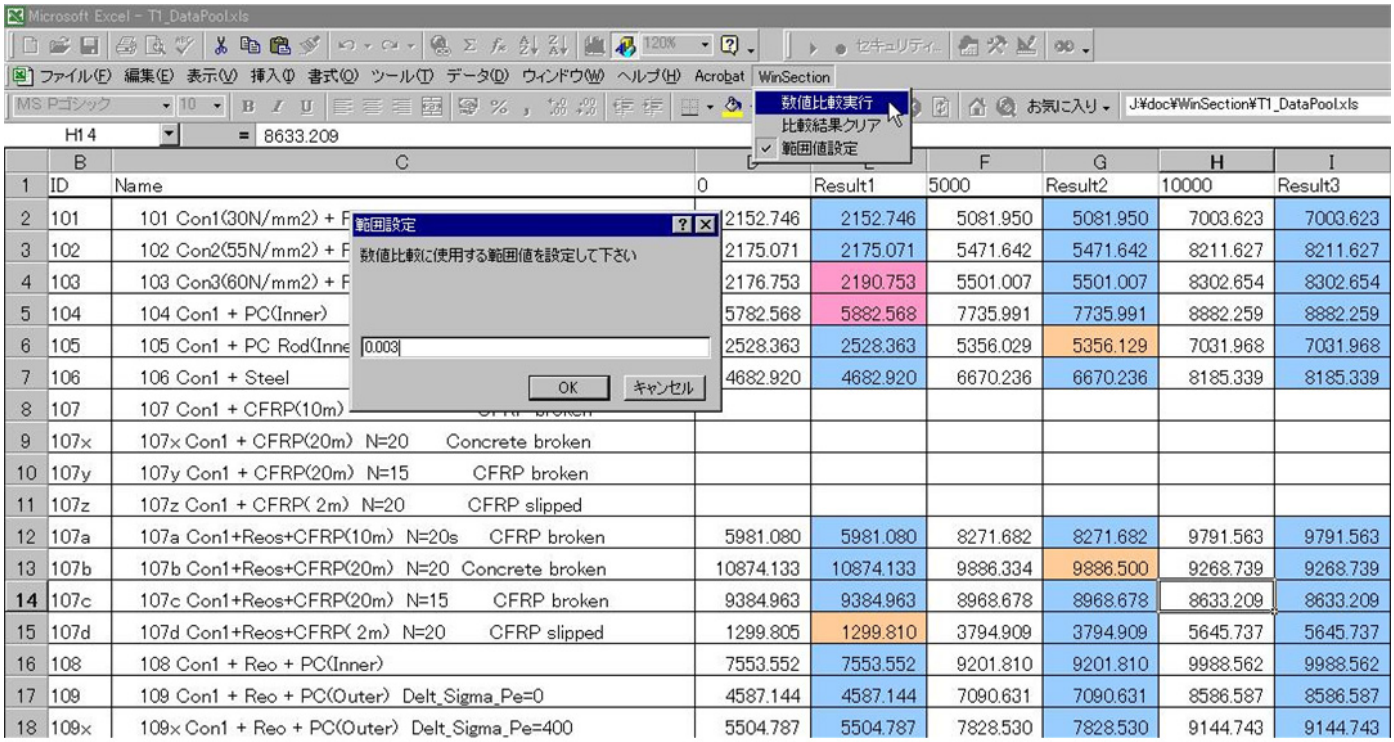


図5. 結果比較シート

5. テストメンテナンス

本事例では Framework の使用に加え,GUIMap を極力使用しないことにより,テストメンテナンスに要する工数を軽減することに留意した．これは自動テストのメンテナンスにおいて，GUIMap の修正に大きな工数がかかることを過去に経験していたためである．具体的な手法としては，AUT の各コントロールに対する制御は可能な限り物理記述を使用するようデザインした．また物理記述に必要な変数(ボタン名称等)は Framework の機能を生かし,ExcelWorkSheet から取得している．
物理記述の例を図 6 に示す．

これにより，メンテナンス発生時の作業のほとんどは Framework で使用する ExcelWorkSheet の修正をすれば良いことになる．

本事例で発生した実際のメンテナンス作業としては，AUT の日本語版インタフェースがリリースされた時にテストセットを新たに作成したが，英語版で使用した WorkSheet を日本語に置き換えるだけで済んだ為，ほぼ一人日で完了した事例がある．
日英各インタフェース用の WorkSheet を図 7 に示す．

6. むすび

本事例で紹介するようなテストセットを構築する場合，一つの障害として初期投資の大きさがあげられよう．物理記述を多用して共通に使用出来る関数を作成しておく手法は，GUIMap に依存する方法と比較すると初期工数が明らかに増大する．ある意味，後々のメンテナンス工数と初期工数とのトレードオフと考えることも出来る．

しかしながら構築したテストの Primitive な部分(共通関数部分)は他のまったく異なるテストにも使用できる可能性が高いこと，テストスクリプトのロジックを追いかけながらメンテナンスする必要がないこと，更には自動テスト自体がメンテナンスを行いながらも長期間に渡って使用しなければ元の取りにくい存在であることを考えれば，検討に値する手法ではないかと考える．

文　献

[1] EMOS Framwwork for WinRunner
http://groups.yahoo.com/group/EMOS_frame/

[2] EMOS Framework Basics 1.ppt
http://www.emos.de/emos/homepage/content.nsf/downlFRMslides?OpenPage

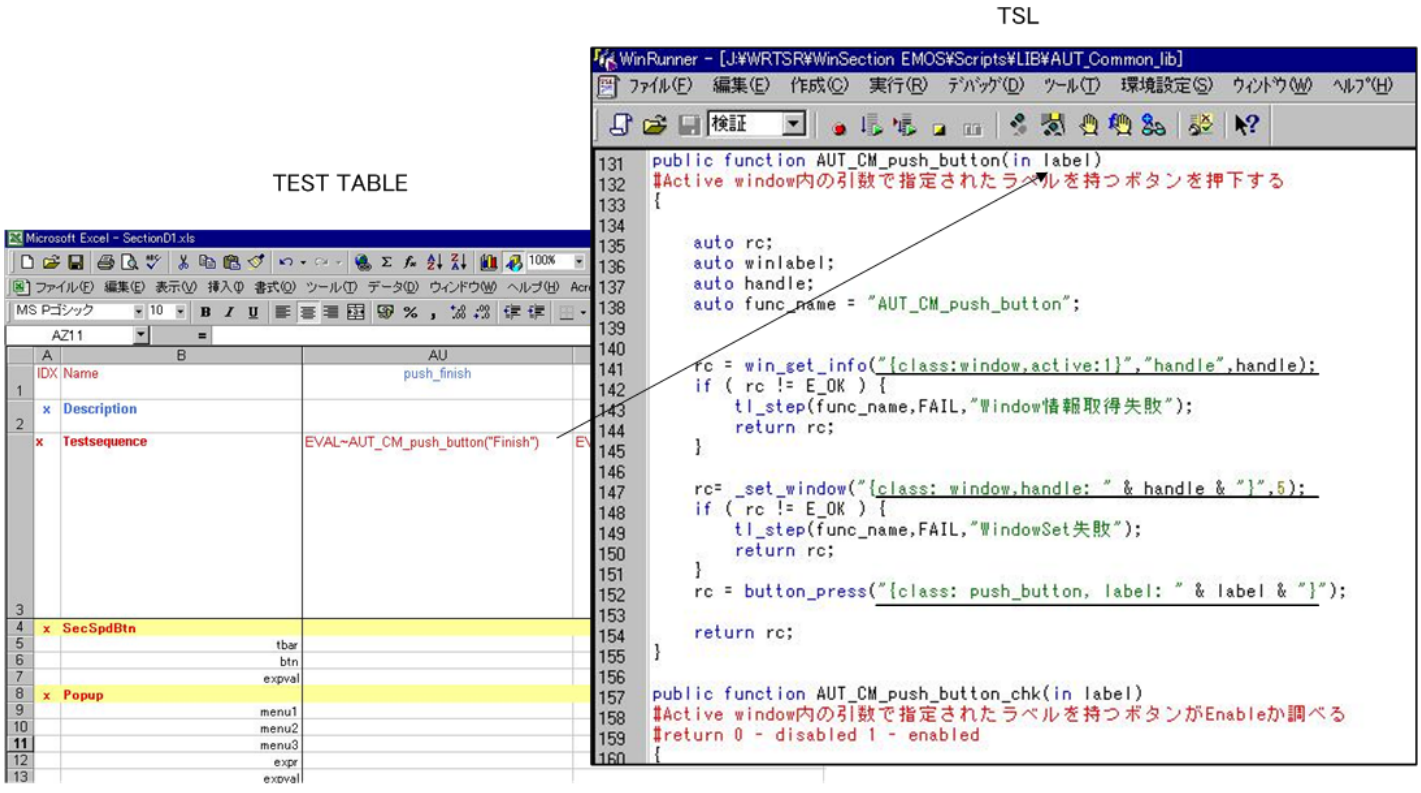
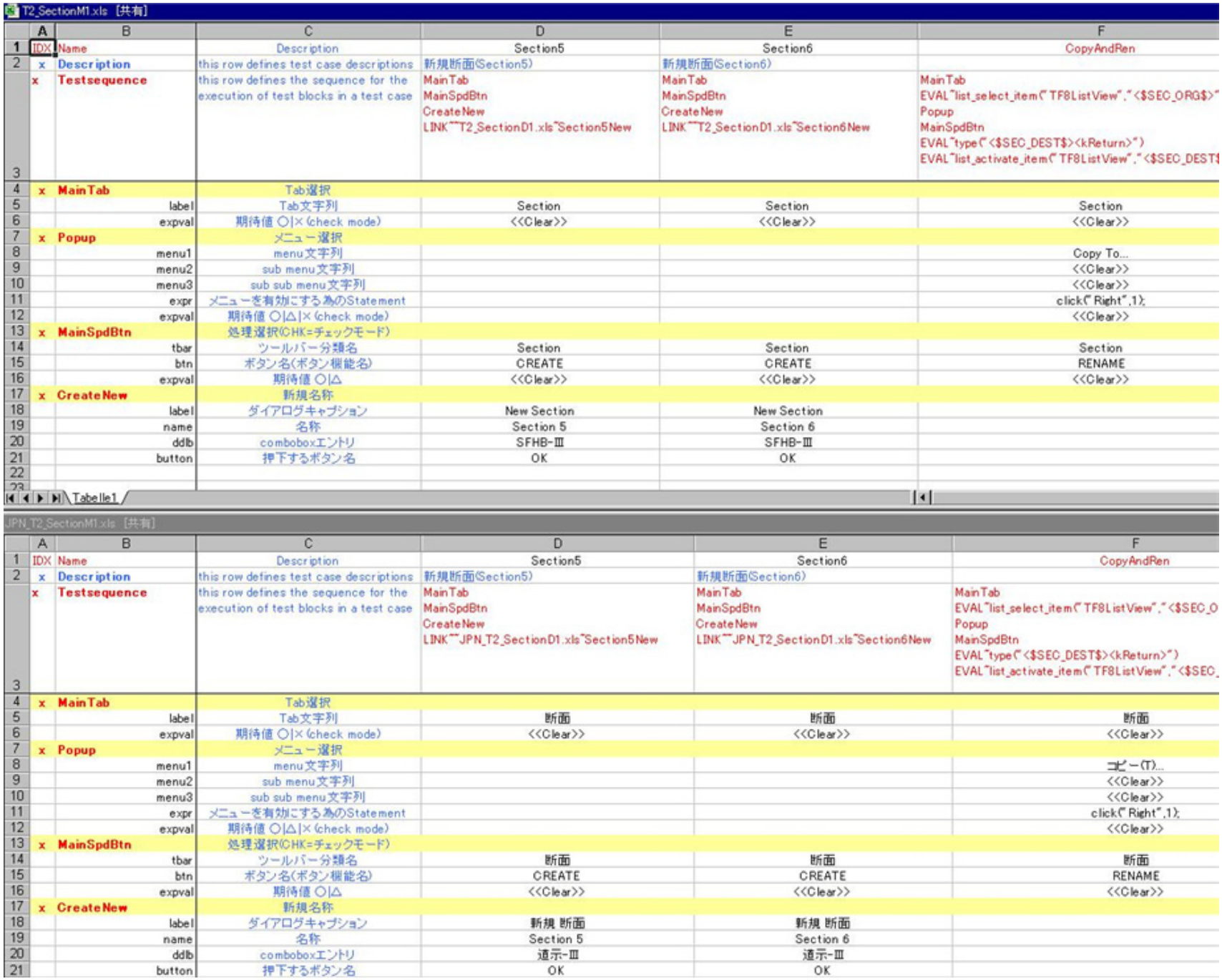


図6. 物理記述例



T2_SectionM1.xls [共有]

	A	B	C	D	E	F
1	IDX	Name	Description	Section5	Section6	CopyAndRen
2	x	Description	this row defines test case descriptions	新規断面(Section5)	新規断面(Section6)	
3	x	Testsequence	this row defines the sequence for the execution of test blocks in a test case	MainTab MainSpdBtn CreateNew LINK""T2_SectionD1.xls"Section5New	MainTab MainSpdBtn CreateNew LINK""T2_SectionD1.xls"Section6New	MainTab EVAL"list_select_item(" TF8ListView","<SEC_ORG>" Popup MainSpdBtn EVAL"type(" <SEC_DEST><kReturn>") EVAL"list_activate_item(" TF8ListView","<$SEC_DEST
4	x	MainTab	Tab選択			
5		label	Tab文字列	Section	Section	Section
6		expval	期待値 ○|× (check mode)	<<Clear>>	<<Clear>>	<<Clear>>
7	x	Popup	メニュー選択			
8		menu1	menu文字列			Copy To...
9		menu2	sub menu文字列			<<Clear>>
10		menu3	sub sub menu文字列			<<Clear>>
11		expr	メニューを有効にする為のStatement			click(" Right",1);
12		expval	期待値 ○|△|× (check mode)			<<Clear>>
13	x	MainSpdBtn	処理選択(CHK=チェックモード)			
14		tbar	ツールバー分類名	Section	Section	Section
15		btn	ボタン名(ボタン機能名)	CREATE	CREATE	RENAME
16		expval	期待値 ○|△	<<Clear>>	<<Clear>>	<<Clear>>
17	x	CreateNew	新規名称			
18		label	ダイアログキャプション	New Section	New Section	
19		name	名称	Section 5	Section 6	
20		ddlb	comboboxエントリ	SFHB-Ⅲ	SFHB-Ⅲ	
21		button	押下するボタン名	OK	OK	
22						
23						

Tabelle1

JPN_T2_SectionM1.xls [共有]

	A	B	C	D	E	F
1	IDX	Name	Description	Section5	Section6	CopyAndRen
2	x	Description	this row defines test case descriptions	新規断面(Section5)	新規断面(Section6)	
3	x	Testsequence	this row defines the sequence for the execution of test blocks in a test case	MainTab MainSpdBtn CreateNew LINK""JPN_T2_SectionD1.xls"Section5New	MainTab MainSpdBtn CreateNew LINK""JPN_T2_SectionD1.xls"Section6New	MainTab EVAL"list_select_item(" TF8ListView","<$SEC_O Popup MainSpdBtn EVAL"type(" <SEC_DEST><kReturn>") EVAL"list_activate_item(" TF8ListView","<$SEC_
4	x	MainTab	Tab選択			
5		label	Tab文字列	断面	断面	断面
6		expval	期待値 ○|× (check mode)	<<Clear>>	<<Clear>>	<<Clear>>
7	x	Popup	メニュー選択			
8		menu1	menu文字列			コピー(T)...
9		menu2	sub menu文字列			<<Clear>>
10		menu3	sub sub menu文字列			<<Clear>>
11		expr	メニューを有効にする為のStatement			click(" Right",1);
12		expval	期待値 ○|△|× (check mode)			<<Clear>>
13	x	MainSpdBtn	処理選択(CHK=チェックモード)			
14		tbar	ツールバー分類名	断面	断面	断面
15		btn	ボタン名(ボタン機能名)	CREATE	CREATE	RENAME
16		expval	期待値 ○|△	<<Clear>>	<<Clear>>	<<Clear>>
17	x	CreateNew	新規名称			
18		label	ダイアログキャプション	新規 断面	新規 断面	
19		name	名称	Section 5	Section 6	
20		ddlb	comboboxエントリ	道示-Ⅲ	道示-Ⅲ	
21		button	押下するボタン名	OK	OK	

図7. 日英両インタフェースの比較