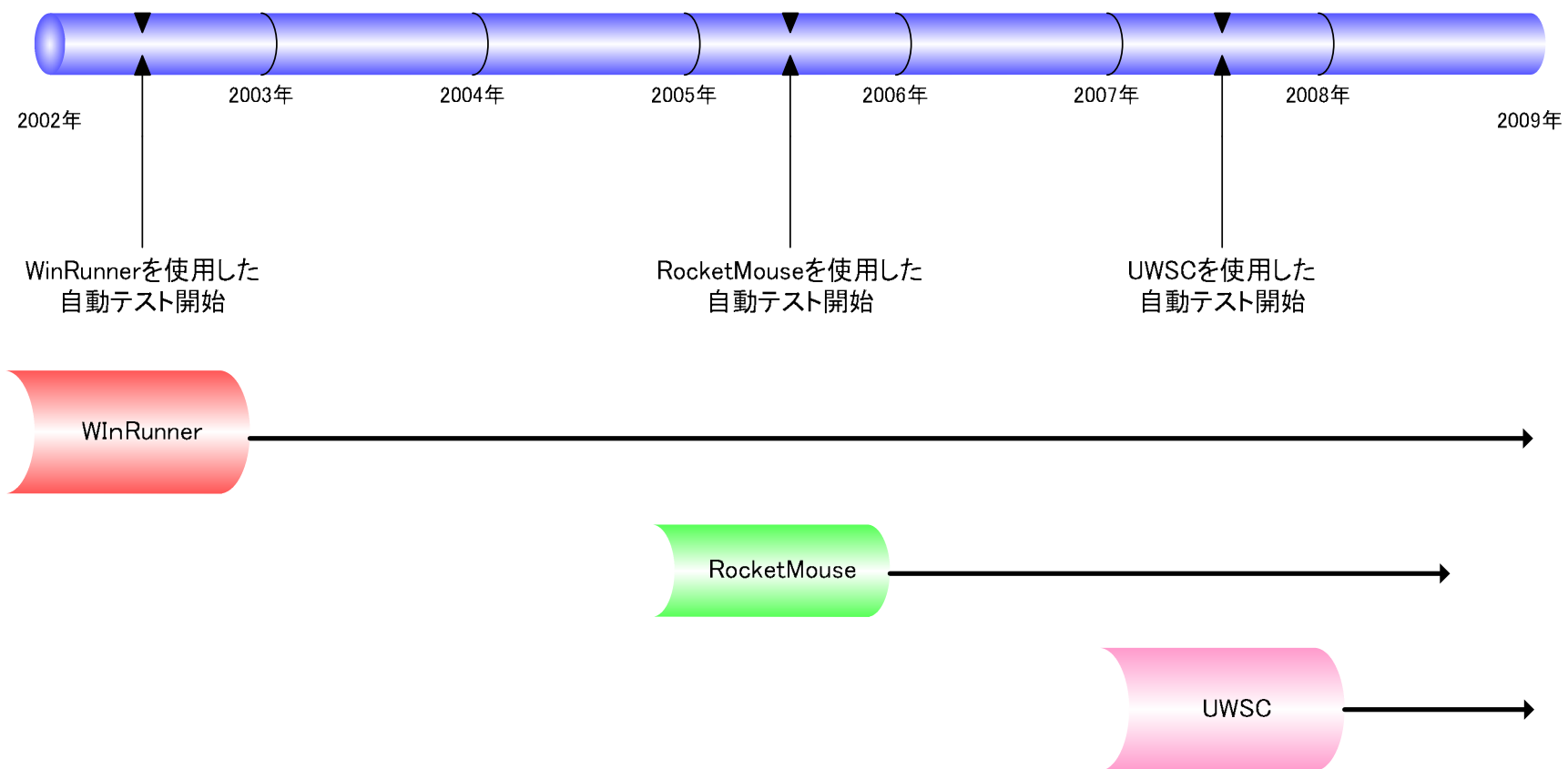


フリーウェアで作る自動テスト
－ UWSCの利用例 －

FORUM8

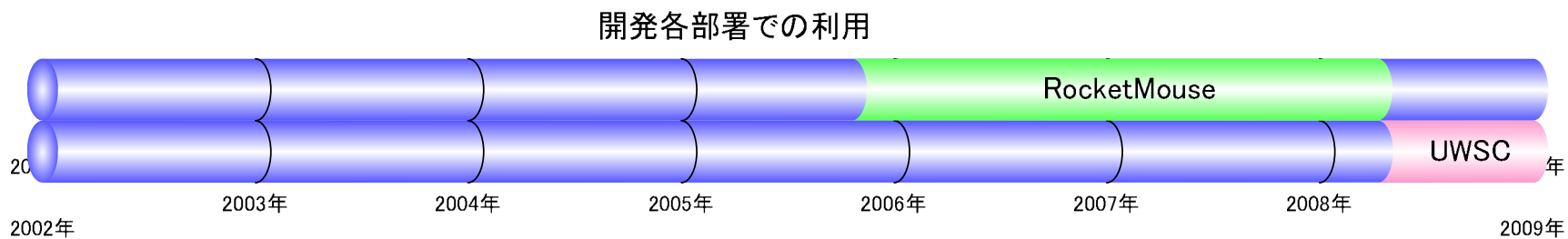
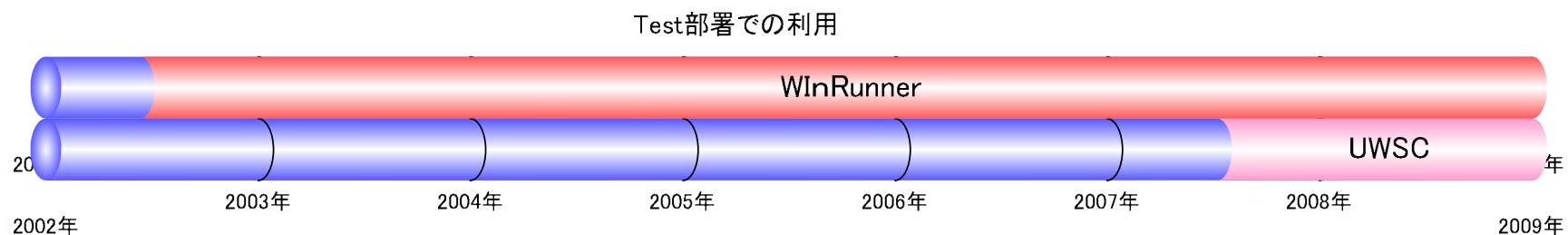
フリーウェアで作る自動テスト -UWSCの利用例-

弊社の自動テストツール使用の歴史



フリーウェアで作る自動テスト -UWSCの利用例-

弊社の自動テストツール使用の歴史(部署別)



フリーウェアで作る自動テスト -UWSCの利用例-

各テストツールの特徴と主たる用途

WinRunner

- ・自動テストツール
- ・スクリプトベース
- ・高価
- ・主に自動回帰テスト用にテスト部署のみで利用

RocketMouse

- ・Windows操作自動化ツール
- ・プログラミング不要
- ・Shareware
- ・簡易自動回帰テスト用にテスト部署で作成、
実行ファイル(Executable)を開発各部署に配布

UWSC

- ・Windows操作自動化ツール
- ・スクリプトベース
- ・Freeware / Shareware
- ・上記2製品のテストをUWSCでリプレース
(RocketMouseは全て。WinRunnerは一部)
- ・新規作成する自動テストは基本UWSCで作成

フリーウェアで作る自動テスト -UWSCの利用例-

UWSC導入による問題点の解決

WinRunner

- ・高価
 - > 開発各部署への導入が困難
- ・多機能
 - > 一定のスキルを要する
- ・重い
 - > 手動では出現することのないエラーの発生
自動テスト運転時間の増大

RocketMouse

- ・基本アナログベースでの記録再生ソフト
- ・スクリプトベースでない
 - > メンテナンスが困難

UWSC

- ・安価
 - > 開発各部署への導入が容易(コストノーリスク)
- ・スクリプトベースで機能(関数)は少なめ
 - > プログラマであれば習熟は容易
- ・軽い
 - > 自動テスト運転時間短縮

フリーウェアで作る自動テスト -UWSCの利用例-

WinRunnerとUWSCの比較

	UWSC	WinRunner	
価格	Freeware (Share版有)	約100万 (+20万/Year)	
IDE	無し	有り	
性格	Windows自動化ソフト	企業向け機能テストツール	
関数の数	およそ80個	およそ670個	
関数例	下記はListView, ListBoxの情報取得、操作に使用する関数の一覧		
	GETITEM CLKITEM STATUS GETSLCTLIST	list_activate_item list_check_info list_check_item list_check_multi_selection list_check_row_num list_check_selected list_check_selection list_collapse_item list_deselect_item list_deselect_range	list_drag_item list_drop_on_item list_expand_item list_extend_item list_extend_multi_items list_extend_range list_get_checked_items list_get_column_header list_get_info list_get_item_info
			list_get_item_num list_get_items_count list_get_item list_get_multi_selected list_get_selected list_get_subitem list_rename_item list_select_item list_select_multi_items list_select_range list_set_item_stat list_wait_info

フリーウェアで作る自動テスト -UWSCの利用例-

WinRunnerを基準とした際のUWSCの優劣

利益

- ・導入及び習得が容易
- ・環境構築が容易
- ・関数が少ない。ある程度最適化された関数の動作
例: ファイルメニュー内の「開く(O)...」サブメニューを選択する関数
UWSC `CLKITEM(id, "開く", CLK_MENU)`
WinRunner `menu_select_item("{class: ¥"menu_item¥", label: ¥"開く(O)... Ctrl+O¥",
parent: ¥"[class: ¥¥¥"menu_item¥¥¥", label: ¥¥¥"ファイル(F)¥¥¥"}¥");`
(*WinRunnerの記述はGUIMAPを使用しない場合)
- ・軽い
 - ・OSへの負荷が少なく、テストオブジェクトの動作を妨げにくい。
 - ・操作(命令)完了待ちの為にタイムラグが小さくて済む為、テスト実行時間の短縮につながる
- ・柔軟性が高い
 - ・Win32API や COMオブジェクトの使用が容易

不利益

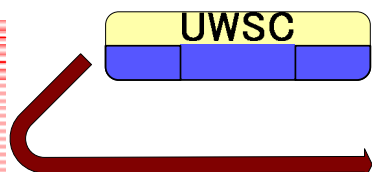
- ・テストオブジェクトの操作上必要不可欠な関数が存在しない場合もある
 - ・例: 任意の位置にある文字列をクリックする関数(win_click_on_text) が無い
-> 別途、DLLを作成した

フリーウェアで作る自動テスト -UWSCの利用例-

自動回帰テストの流れ

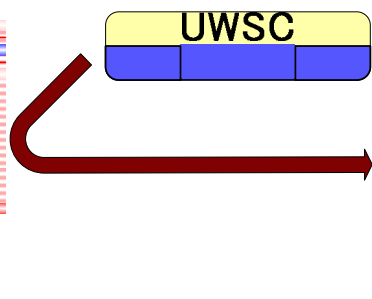
1. 旧バージョンテストオブジェクトの結果をキャプチャ

テストオブジェクト
テスト対象アプリ
(古いバージョン)



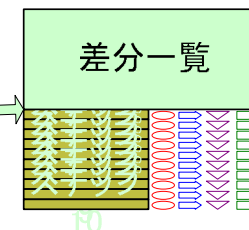
2. 新バージョンテストオブジェクトの結果をキャプチャ

テストオブジェクト
テスト対象アプリ
(新しいバージョン)

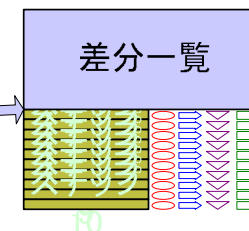


3. キャプチャファイル比較

画像
差分検出ツール
(*1)



テキスト
差分検出ツール
(*2)



(*1) 独自開発ツール使用 (ExcelVBA+Delphi) <http://www.vector.co.jp/soft/winnt/prog/se347531.html>

(*2) WinMerge等のフリーウェア <http://www.geocities.co.jp/SiliconValley-SanJose/8165/winmerge.html>

フリーウェアで作る自動テスト -UWSCの利用例-

UWSCのインストールと環境構築

Download、インストール

- http://www.uwsc.info/download.html
- フリーウェア版は任意のディレクトリに解凍
- シェアウェア版はインストーラを実行

環境構築

- IDEを持たないので任意のエディタを必要に応じてカスタマイズ。
お勧めはサクラエディタ。
- DL:
http://members.at.infoseek.co.jp/sakura_editor/index.html
- カスタマイズ解説:
<http://makotowatana.ld.infoseek.co.jp/program.html#UWSC>
上記URLを参考に強調キーワード、外部HTML、外部マクロ等を設定

カスタマイズ後のサクラエディタ

```
5 |  
6 | dim gPosX, gPosY, gPosW, gPosH, id1, icnt_entry, inum_entry; |  
7 | |  
8 | call RegressionTestFunction.uws; | //サブルーチン読込 |  
9 | print "====テスト開始===="; |  
10 | |  
11 | id1 = GETID(GET_THISUWSC_WIN); |  
12 | ACW(id1, 0, 0); |  
13 | id1 = GETID(GET_LOGPRINT_WIN); |  
14 | ACW(id1, gLPosX, gLPosY, gLPosW, gLPosH); | //ログWindowの位置とサイズ |  
15 | //ファイル毎Loop===== |  
16 | for icnt_entry = 0 to inum_entry - 1 | //パラメータLoop |  
17 | |  
18 | next |  
19 | exit; |
```

UWSCスクリプト解説

//起動

```
rtn = EXEC("D:¥Program Files¥FORUM 8¥UC-BRIDGE6¥UC_BRIDGE4.exe");
```

//WindowID取得

///タイトル文字列の一部でもOK

///複数HITする場合は直近でACTIVEだった方のWindowを返してくる

```
id_main = GETID("UC-BRIDGE");
```

//メニュー選択

///ユニークな文字列であればサブメニューの文字列だけでOK

///末尾のニーモニックは不要

///FULLPATHで指定する場合は¥サインでつなぐ("ファイル¥開く")

```
CLKITEM(id_main, "開く", CLK_MENU);
```

//ファイル読込

///操作したいウィンドウに対してはその都度IDを取得(ウィンドウが消滅するまでIDは使える)

```
id_open = GETID("ファイルを開く");
```

///id内のエディットコントロールに文字列を送る。3番目の引数は左上から数えた順番

```
SENDSTR(id_open, "D:¥Program Files¥FORUM 8¥UC-BRIDGE6¥Data¥PC_BEAM.f8q", 1, TRUE);
```

///プッシュボタン、チェックボタン、オプションボタン等すべてのボタン類はCLK_BTNに分類

///複数の同一名称がある場合は同一名称のボタンの左上から数えた順番を指定

```
CLKITEM(id_open, "開く", CLK_BTN);
```

UWSCスクリプト解説

//計算実行

```
CLKITEM(id_main, "セット計算", CLK_MENU);
```

```
id_calc = GETID("セット計算");
```

```
CLKITEM(id_calc, "実行", CLK_BTN);
```

//結果画面に移動

```
CLKITEM(id_main, "結果確認", CLK_TAB);
```

```
///ツリービュー内の指定方法はメニューと同じ(FULLPATH指定の場合は¥でつなぐ)
```

```
CLKITEM(id_main, "変位", CLK_TREEVIEW);
```

//結果画面をキャプチャしてファイル出力

```
///ウィンドウ内のコントロールのウィンドウハンドルを取得
```

```
hwnd = GETCTLHND(id_main, "TPaintPanel");
```

```
///ハンドルをIDに変換
```

```
id_paint = HNDTOID(hwnd);
```

```
///BMP形式でキャプチャ
```

```
SAVEIMG("test.bmp", id_paint);
```

UWSCスクリプト解説

```
//計算書出力
CLKITEM(id_main, "全データ", CLK_MENU);
///計算書出力編集ツール表示
id_f8 = GETID("F8出力");
///メニューをCLKITEMで選んでも良いがここではショートカットキーのCTRL-Sで
///保存Dialogを呼び出す
SCKEY(id_f8, VK_CTRL, S);
id_save = GETID("名前を付けて保存");
///ファイル名を入力して保存ボタンをクリック
SENDSTR(id_save, GET_CUR_DIR + "¥test.txt", 1, TRUE);
CLKITEM(id_save, "保存", CLK_BTN);
///計算書出力編集ツールを閉じる
CLKITEM(id_f8, "閉じる", CLK_MENU);

//終了
///メニューを選択しても良いが、CTRLWINでも終了する
CTRLWIN(id_main, CLOSE);

exitexit;
```

フリーウェアで作る自動テスト -UWSCの利用例-

Win32APIの利用

```
// アクティブウィンドウの左上に時間を表示(デスクトップに文字を直書き)
DEF_DLL GetDC(hwnd): hwnd: USER32
DEF_DLL ReleaseDC(hwnd, hwnd): int: USER32
DEF_DLL TextOutW(hwnd, int, int, wstring, int): Bool: GDI32

dc = GetDC(0) // デスクトップのDC
while ! GETKEYSTATE(VK_ESC) // ESCキーにて終了
  w = GETID(GET_ACTIVE_WIN)
  GetTime()
  s = G_TIME_HH2+":"+G_TIME_NN2+":"+G_TIME_SS2
  TextOutW(dc, Status(w,ST_X)+2, Status(w,ST_Y)+2, s, Length(s))
  Sleep(0.5)
wend
ReleaseDC(0, DC)
```

UWSC ホームページのサンプルより

自前で作成したDLLも使用可能

```
http://f30.aaa.livedoor.jp/~seiten/dllcreate.htm
UWSC用DLL作成解説サイト(Borland C + BCC Developer)
```

COM オブジェクトの利用

```
// FileSystemObjectオブジェクトを利用してフォルダを新規作成
dim str1 = GET_CURRENT_DIR + "¥capture¥";
try
    dim objWf = CREATEOLEOBJ("Scripting.FileSystemObject");
    Ifb objWf.FolderExists(str1) = FALSE then
        objWf.CreateFolder(str1);
    endif
except
    PRINT("OLEエラー");
endtry
```

```
// Excelファイルのセル内容読み出し
dim objExcel= CREATEOLEOBJ("Excel.Application");
objExcel.Visible = TRUE;
objExcel.WorkBooks.Open(FileName := GET_CURRENT_DIR + "¥param.xls");
dim str_flg = objExcel.ActiveSheet.Range("A1").Value;
objExcel.Quit;
```