

## Framework及びExcelを使用した自動テスト効率化試行

### (1)自動テストの問題点

- ・自動テストツールの使用経験から提起された問題点と改善策

### (2)EMOS Framework概要

- ・問題点改善策の一つであるFrameworkの概要

### (3)テスト事例紹介

- ・概要
- ・自動テスト実施
- ・テスト結果比較(問題点改善策の一つであるExcelを使用した結果比較)

### (4)メンテナンス工数の削減

- ・本テスト事例におけるメンテナンス工数削減へのアプローチ

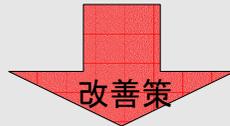
### (5)考察

- ・本テスト事例で採用した手法のメリットとデメリット

## 自動テストツールの問題点

### メンテナンス工数

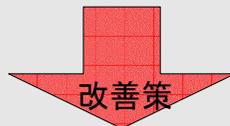
- ・テストスクリプト及びGUIマップの変更に工数を要する
  - 特にGUIマップの変更に時間と根気を要するケースが多い
- ・AUTのマルチリンガル化への対応が困難



### Frameworkの採用

### 数値結果比較 (表形式の大量の数値を比較する場合)

- ・BitMap比較はコーディングは容易であるが、結果比較が硬直的
- ・閾値(誤差)許容の設定が困難



### Excel(VBA)との関係

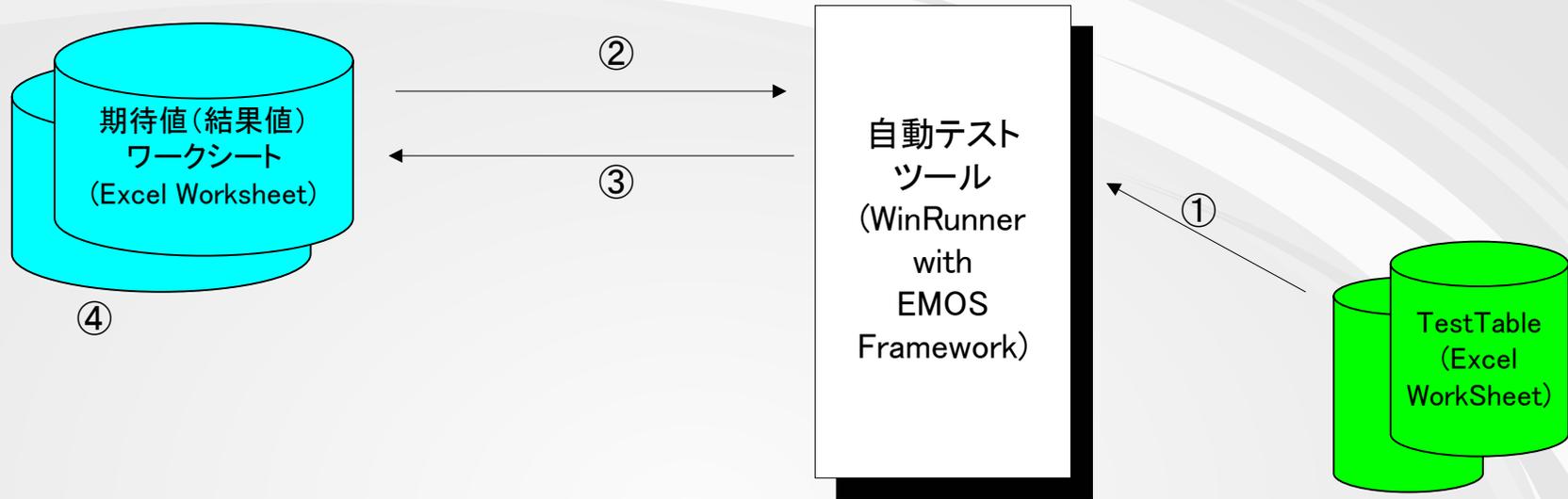
| OK            | 曲げ耐力                   |                                 |
|---------------|------------------------|---------------------------------|
|               | Mc < Mu (kNm)          | Mumin < M < Mumax (kNm)         |
| 仮想部材 1        | 883.876 < 4682.920 OK  | -4682.920 < 0.000 < 4682.920 OK |
| Check Point 1 | 883.876 < 4682.920 OK  | -4682.920 < 0.000 < 4682.920 OK |
| - User Case 1 | 883.876 < 4682.920 OK  | -4682.920 < 0.000 < 4682.920 OK |
| - User Case 2 | 2082.988 < 6670.236 OK | -6670.236 < 0.000 < 6670.236 OK |
| - User Case 3 | 3282.099 < 8185.339 OK | -8185.339 < 0.000 < 8185.339 OK |
| - User Case 4 | 0.000 < 156.361 OK     | -156.361 < 0.000 < 156.361 OK   |
| - User Case 5 | 0.000 < 82.842 OK      | -82.842 < 0.000 < 82.842 OK     |
| - User Case 6 | 883.876 < 4682.920 OK  | -4682.920 < 0.000 < 4682.920 OK |

各行の3種類の数値が各々正しいか検証する

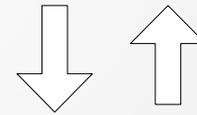
## EMOS FRAMEWORKの導入

### Excel Worksheet(TestTable)を用いた自動テストの制御

- ・自動テストのテストフローをWorksheetに記述
  - －記述された関数を順序良く呼び出す
- ・関数への引数をWorksheetから取得
  - －呼び出した関数にパラメータを与える



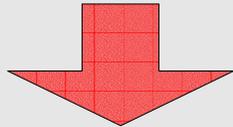
- ①自動テスト起動後はTestTableの記述に従い、AUT自動実行
- ②計算結果を取得すべき断面の情報を取得
- ③計算結果を取得し、結果を書き戻す
- ④結果値の整形



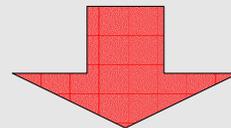
## メンテナンス工数削減の為の手法

### GUIマップからの独立

- ・物理記述を多用
- ・物理記述で使用するパラメータはFrameworkのTestTableより取得



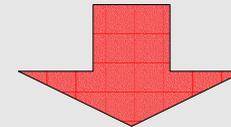
- ・メンテナンス対象の主体がテストスクリプト及びGUIマップからTestTableに移行



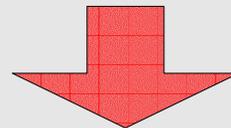
- ・メンテナンスの単純化

### スクリプトのメンテナンスフリー化

- ・共通関数群としてデザイン
- ・フローはTestTableに依存



- ・メンテナンス対象の主体がテストスクリプト及びGUIマップからTestTableに移行



- ・メンテナンスの単純化

## 考 察

### メリット

- メンテナンスの単純化
- メンテナンス時工数減
- マルチリンガルインタフェースへの対応が容易

### デメリット

- 初期工数の増大
- 共通関数群の作成
- Framework自体の習得時間