ソフトウェアテストシンポジウム(JaSST)2007

ソフトウェア国際化テスト技法
－擬似翻訳手法の改良－

日本アイ･ビー･エム　ソフトウェア開発研究所
青山 希、　田中 祐子

© Copyright IBM Corporation 2007

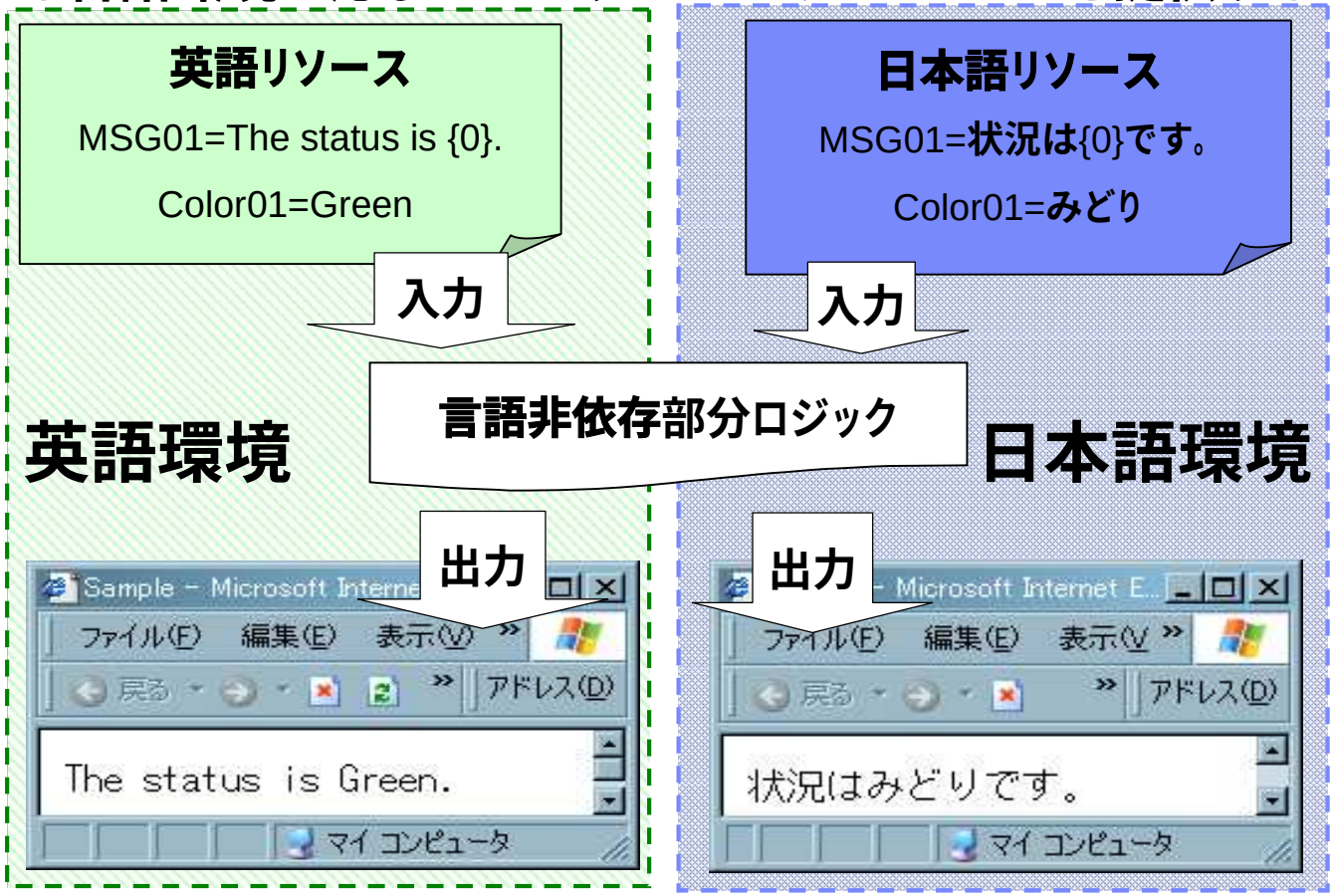
概要

- ソフトウェア国際化テストの位置づけと既存手法の紹介
- 国際化テストにおける問題検出の精度を上げるための擬似翻訳手法の改良
  - 国際化テストで使用する擬似翻訳手法の改良とツールの実装を行った
  - 従来手法では検証が不可能な項目についても検証が可能になった
  - 実際の開発プロジェクトに適用し効果を検証した

© Copyright IBM Corporation 2006

国際化されたソフトウェア

- 言語非依存ロジック部分と、言語に依存するリソース・ファイルを分離
- 実行する言語環境に応じたユーザー・インターフェースを提供する



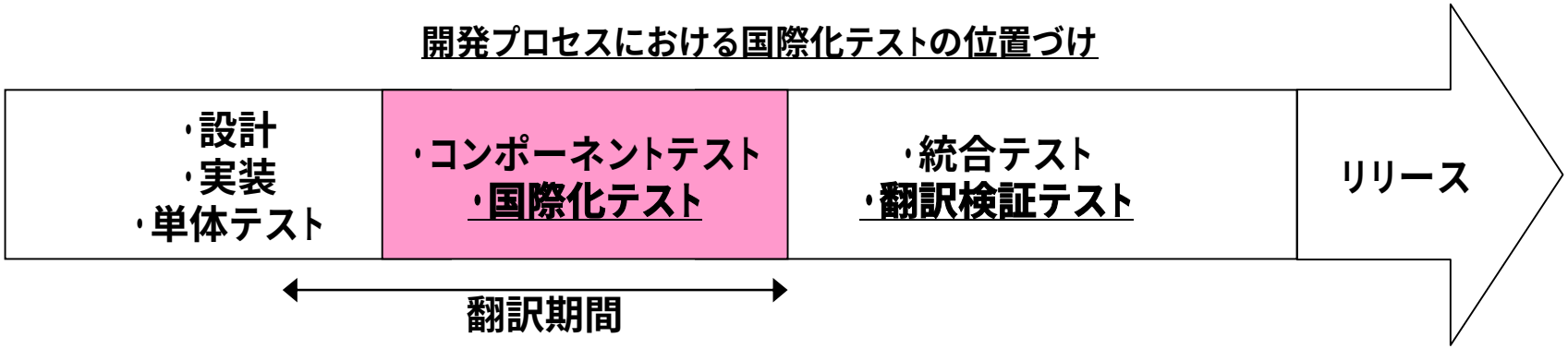
© Copyright IBM Corporation 2006

ソフトウェアの国際化テスト

- ソフトウェアの国際化に関する機能を検証するテスト
- 製品・コンポーネント開発に組み込まれている開発プロセスのひとつで翻訳検証テストに先立って行われる
- 以下のような国際化機能の検証を行う
  - 文字が正しく翻訳され表示され、ソートされるか
  - サポートするどの言語環境でも正しく動作するか
  - 多言語データを取り扱うことができるか
  - 数値、カレンダー、通貨などの各言語習慣に沿って扱うことができるか

本発表で扱う検証項目

開発プロセスにおける国際化テストの位置づけ



© Copyright IBM Corporation 2006

従来手法

- 言語リソースファイルの翻訳を擬似的に行い、翻訳完了前の国際化テストを可能にする
  - 元の英語文字列に問題を引き起こしやすい各国語文字列を追加する

Rational Pseudo Translation toolによる擬似翻訳例

擬似翻訳されたリソースファイル

```
MSG01=[表竹～The status is {0}. ^^^^ja]
Color01=[表竹～ Green ^^^^ja]
Color02=[表竹～ Red ^^^^ja]
Color03=[表竹～ Yellow ^^^^ja]
MSG02=[表竹～ Refresh the status ^^^^ja]
```

正しくリソースが外部化されている

文字化けしている
ハードコードされている

© Copyright IBM Corporation 2006

従来手法の課題

擬似翻訳されたリソースファイル

MSG01=[表竹～The status is {0}. ^^^^ja]

Color01=[表竹～ Green ^^^^ja]

Color02=[表竹～ Red ^^^^ja]

Color03=[表竹～ Yellow ^^^^ja]

テスターが確認する画面

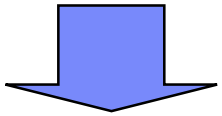
- 検証項目1：画面には“Green”が代入文字列であることが示されないため
  ハードコードされていることが判別できない
- 検証項目2：同じ文字が先頭に追加されるため英語と同じ順番にしかソートされない
- 検証項目3：表示を検証できる文字種が数種類に限定される。

© Copyright IBM Corporation 2006

提案する手法

- 擬似翻訳リソースの変換手法を変更
  - 従来の変換手法

> •文字列の先頭と末尾にあらかじめ定義された日本語文字を追加



  - 新たに提案する変換手法

> •先頭文字の擬似翻訳
>
> 検証項目2「ソート結果の妥当性検査」,
> 検証項目3「言語固有文字群の表示可能性検証」の問題を解決
>
> •可変文字列の代入を明示するマーカーの設定
>
> 検証項目1「ハードコード文字列の検出」の問題を解決

© Copyright IBM Corporation 2006

先頭文字の擬似翻訳

- 変換手法：各アルファベットを変換するためのテーブルをもとに、先頭文字の前後にそれぞれ異なる文字を追加する
  - 変換前の英語リソース

```
Color01=Green
Color02=Red
Color03=Yellow
```

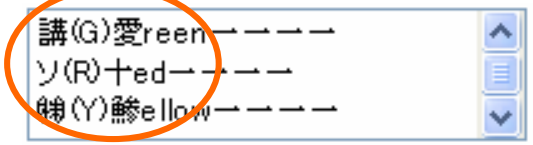
  - 変換後の擬似翻訳リソース

```
Color01=講(G)愛reenーーーーー
Color02=ソ(R)十edーーーーー
Color03=㈱(Y)鯵ellowーーーーー
```

- 適用例

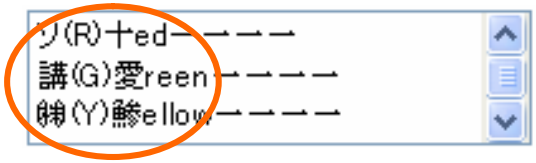
さまざまな文字について表示の検証をすることが可能

  - 国際化に問題があるソフトウェアの場合



期待される順序で
ソートされていない

  - 正しく国際化されたソフトウェアの場合



言語環境に適した順序で
ソートが行われる

© Copyright IBM Corporation 2006

可変文字列の代入を明示するマーカーの設定

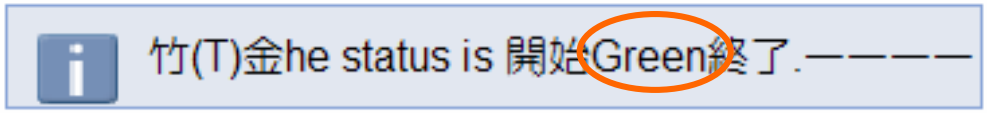
- **変換手法: 代入位置を示す{ } に「開始」「終了」を追加する**
  - **変換前の英語リソース**

```
MSG01=The status is {0}.
```

  - **変換後の擬似翻訳リソース**

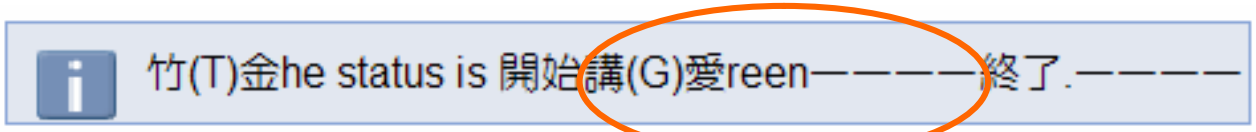
```
MSG01=竹(T)金he status is 開始{0}終了.ーーーーー
```

- **適用例**
  - **文字列がハードコードされているソフトウェアの場合**



“Green”**が**
ハードコードされている

  - **正しく国際化されたソフトウェアの場合**

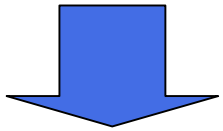


“Green”**が**
外部化されている

© Copyright IBM Corporation 2006

期待される効果

- 擬似翻訳手法の改良で国際化テストの問題検出精度があがる
  - 可変文字列のハードコードが検出できる
  - ソート結果の検証が可能になる
  - ひとつのリソースファイルで、様々な文字の表示も検証することができる



- 国際化テストそのものの品質向上
  - 翻訳検証テストで発見される国際化品質に関する問題が減少する

  →工数の削減

© Copyright IBM Corporation 2006

検証結果

- **実際の開発プロジェクトへの適用**

  参考:リソースファイルに含まれるメッセージの数 → 約2000

  - 本手法でなければ発見することができない
    ハードコードに関する問題を数件発見することができた
    参考:国際化テストで発見された問題 → 全体で約30件
  - テーブルのソートに関して日本語環境で
    正しく動作することを検証することができた
  - 特殊なブラウザ環境では特定の文字の表示で
    文字化けが起こることを発見できた

© Copyright IBM Corporation 2006

まとめ

- 目的
  - 国際化テストの問題検出精度をあげ、国際化テストそのものの品質を向上する
- 提案する手法
  - 擬似翻訳手法の改良
    - 可変文字列のハードコードが検出できる
    - ソート結果の検証が可能になる
    - いままで検証できなかった文字も検証することができる
- 期待される効果
  - 国際化テストの品質向上　　→翻訳検証テストでの手戻りが減少する
- 今後の課題
  - 検証対象の文字種が増えたことで文字化けの判定が難しくなる場合がある
  - アラビア語などBi-Directional言語環境での検証

© Copyright IBM Corporation 2006