World Wide Naviによる
ソフトウェア国際化開発の自動化・効率化

2009年7月1日

国際化JP株式会社

岡村　純一

ソフトウェアの国際化の主なポイント

- シングルソース化（1ニュートラル＋Nローカル）
- Unicode（UTF-8)化
- 文化依存情報の切り替え
- 開発とリソース追加/翻訳の分離
- 言語切り替えの動作確認
- ノウハウの共有

World Wide Naviにより
自動化/効率化が出来る作業

- シングルソース化

  文字列外部化機能によりメッセージやGUI情報の動的切り替えを自動的に挿入

- 非Unicode対応コードの検出

  プラグイン/アドインによりEclipse、Visual Studioから直接行えることで対応効率が上昇

- 文化依存処理の検出

  解析ルールのカスタマイズにより検出パターンを蓄積

- ルールの共有

  ルールのインポート/エクスポートにより解析方法を共有化

今後World Wide Naviで
自動化/効率化が出来る作業

- 国際化と現地語化の分離

  ローカリゼーションツール（Sisulizer）との連携により、開発はマスターリソースのみメンテナンス

- 開発作業と連動した言語切り替え

  切り替え機能を使った開発とテストの同時進行

World Wide Naviの主な特徴

- 主要言語サポート
  (C/C++、Java、C#、VB…)
- 主要IDEとの連携
  (Eclipse、Visual Studio…)
- 主要プラットフォームサポート
  (Windows、Linux….)
- 多項目のソースコード解析
- 有用な文字列の外部化
- 柔軟な解析ルールの編集
- 自身の国際化(英、日、韓、…)

サンプルコードでのデモンストレーション

- Visual Studioとの連携による効率化
- 問題点の抽出の効率化
- 埋め込み文字列の自動外部化
- 外部化されたリソースをバイナリとしてローカライズ（Sisulizerとの連携）

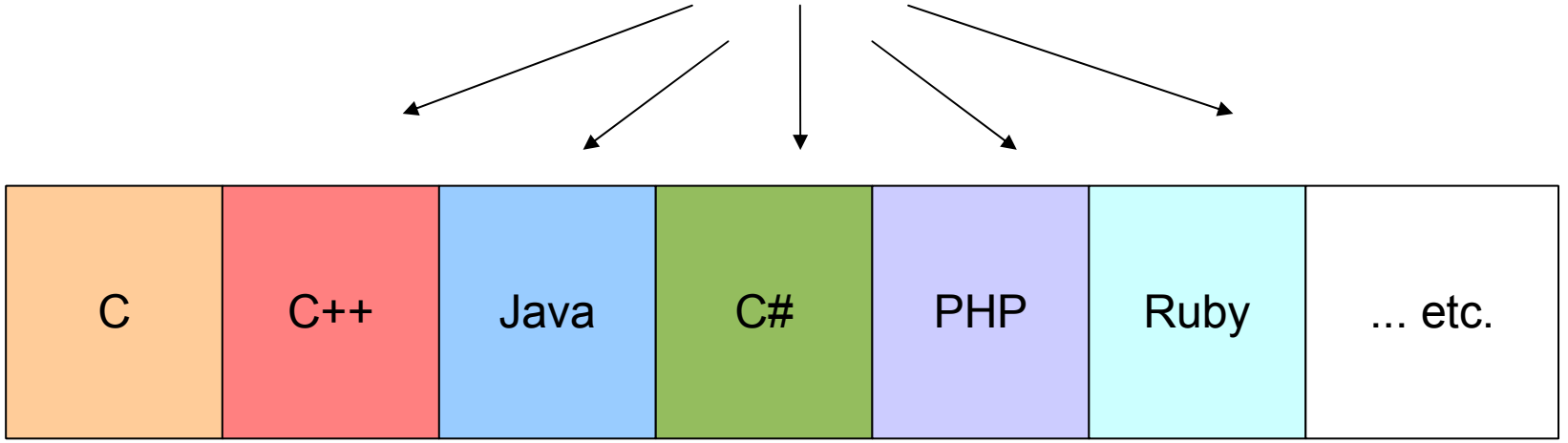
解析ポイント

1. 宣言部
   (char, TCHAR, String...)
2. 関数使用部
   (printf, MessageBox, DateFormat...)
3. 文字列部
   (“Hello, Wolrd”, “submit”, “File”...)
4. 文字比較部
   (if (c > '0')…)
5. 必要キーワード
   (setlocale, loadlibrary...)

解析方法

共通ポイントを各プログラミング言語に適用

1. 宣言部
2. 関数使用部
3. 文字列部
4. 文字比較部
5. 必要キーワード



適用範囲

- 現在はデスクトップ、クライアントが主流
- 今後、モバイル（iPhone、Android、Windows Mobile…）、組み込み（Embedded Linux)、Webアプリケーションへ拡張予定
- 対応OSもMac、Solarisなども予定（複数OS混在環境での統一使用）

情報発信

- 製品ページは随時更新
- ダウンロードユーザーにはリリース通知
- 今後更なる情報発信を企画中

おわり

Q&A