

Internet Week 2010

ユニバーサルコンテンツ時代の文字コード ～ Unicodeがもたらした光と闇～

『国際化プログラミングと文字コード』

2010年11月24日

末廣 陽一
代表取締役
国際化JP株式会社
<http://www.kokusaika.jp/>

1. 自己紹介・会社紹介

自己紹介 (略歴)

1962年3月	原産地: 鹿児島県
1986年3月	九州大学大学院総合理工学研究科情報システム学専攻修士課程修了。英日間の機械翻訳を研究。
1986年4月	日本デジタルイクイップメント株式会社(日本DEC)に入社。日本DEC研究開発センター株式会社に配属。
1998年10月	会社合併により、コンパックコンピュータ株式会社に編入。
2002年11月	会社合併により、日本ヒューレット・パカード株式会社に編入。
2003年12月	日本ヒューレット・パカード株式会社(日本HP)退社。
2004年2月	ビジネスサーチテクノロジー株式会社(BST)を共同で設立。 取締役 研究開発センター ソフトウェア国際化開発部部長に就任。オープンシステム技術開発部部長を兼任。
2006年4月	ビジネスサーチテクノロジー株式会社退社。
2006年6月	国際化JP株式会社を設立。代表取締役に就任。現在に至る。

自己紹介 (国際化についての経歴)

- DEC, Compaq, HPでUNIX OSの国際化開発、開発マネージメント、プロダクトマネージメントを担当。日本のTru64 UNIX OS開発チームのプロジェクトリーダーを約10年務める。HP退職前の1-2年は、USやヨーロッパなど世界中の国際化チームの作業を日本に集約。そのマネージメントを務めた。
- UNIX、国際化の標準化委員で会社代表、日本代表団の一員として活動。X/Open 国際化 WG (DEC代表)、SC22/C Language (JIS、ISO) 委員、SC22/POSIX (JIS、ISO) 委員
- IBM木戸氏、Sun(当時)樋浦氏らとLinux国際化機能の標準化組織LI18NUX(その後、Open18Nに名称変更)を設立。I18N system architecture subgroup のリーダーとして、Linux OSの国際化仕様Li18nux2000をまとめた。AIT 小山教授ともこの時期に出会い、日本のコミュニティメンバーと共に活動した。
- 著書: 共立出版「国際化プログラミング」(エディタとしてまとめた)。Software Design、Computer Todayなどへの記事執筆。
- Compaq/HP時代には社外イベント(Linux World等)や会社主催のシンポジウムなどでソフトウェア国際化についての講演を実施。国際化JP起業後も CICC のイベントなどで国内、国外(タイ、中国、シンガポール、マレーシア、ネパールなど)でソフトウェアの国際化について講演。

著書：国際化プログラミング

- 「国際化プログラミング」
- I18N ハンドブック-

清兼義弘・末廣陽一/編著
共立出版 ISBN4-320-02904-6
1998年8月25日 初版

- 国際化の手法について日本語でまとめた本は他にない。
- 今ではまずありえない高い定価**10,000円**！
- 技術解説部分は内容が古くなっているが、前半のコンセプト部分は国際化の基礎知識満載。



とても大変だけれど、著書を持つことの影響は大きい

ライフワーク：国際化という分野

- 積み重ねてきたもの：
社内での開発、社外での標準化活動、両方で15年以上もソフトウェアの国際化に取り組んできた。
- かなりニッチな世界だが、グローバルで成功している企業とそうでない企業との認識や取り組みの差が大きい分野。
- 欧米のグローバル企業は、過去に長期的視点から多大な投資を行って、現在がある。マイクロソフト、IBM、オラクル、SAP、など。Google の成功もグローバルに展開しているから。
- グローバルビジネスでは国際化が重要。

法人名	国際化JP株式会社 (コクサイカジェーピーカブシキガイシャ)		
代表者名	末廣 陽一		
所在地	東京都千代田区内神田2 - 11 - 6 共同ビル(内神田) 4F		
TEL	03-6206-8523	FAX	03-6206-8524
URL	http://www.kokusaika.jp/		
業種	ソフトウェア開発	設立年月日	2006年6月8日
資本金	950万円	決算月	5月
製品・サービス	ソフトウェア国際化およびウェブサイトの国際化に関するコンサルティング、製品の開発および販売、各種サービスの提供など。		
その他	東京商工会議所 会員番号 C2124653		

検索


ホーム ▶ Info ▶ ソフトウェア国際化の巨星落つ、樋浦秀樹氏がUSで死亡

Japanese ▼

ホーム	コンセプト	製品紹介	サービス	OSS	事例紹介	Info	会社概要
-----	-------	------	------	-----	------	------	------



ニュースリリース

2010年4月16日
国際化JP株式会社

ソフトウェア国際化の巨星落つ、樋浦秀樹氏がUSで死亡

長年にわたってサンマイクロシステムズの国際化(18N)アーキテクトを務めた樋浦秀樹氏が4月7日、アメリカで癌のために亡くなった。樋浦氏は、国際化のテクノロジーの発展と普及振興のために多大な貢献を行った。我々は、彼のコンピュータテクノロジーへの造詣の深さ、卓越した技術力、そして情熱と笑顔を忘れることのない、国際化JPIは、心の底から彼の業績とコンピュータ業界への貢献に敬意を表するものである。

友人として そして同志として
末廣 陽一
国際化JP株式会社 代表取締役

(1報より1週間を経過して掲載しました)

国際化には二通りある

- 外から受け入れる方向（インバウンド）
 - 国外から人材、文化、商品、サービスなどを受け入れる
 - 受け入れた結果生じる変化、あるいは受け入れるために行う変化(準備)が国際化
- 外に出していく方向（アウトバウンド）
 - 国外へ人材、文化、商品、サービスなどを出していく
 - 出した結果生じる変化、あるいは出すために行う変化(準備)が国際化

- ソフトウェアの国際化

- 一つのソースコード、一つのバイナリで世界中で利用できるようにソフトウェアを作る技術

- 自社開発の開発支援ツール『World Wide Navi』、ソースコード診断サービス『ソースコードクリニック』、コンサルティングなどを提供

- ウェブサイトの国際化

- 多言語のページを切り替えて表示したり、ページの作成や管理を容易にする技術

- オープンソースCMSのJoomla!を活用したソリューション JoomWeb を提供

2. ソフトウェアの国際化とは



ソフトウェアの国際化とは

- 国際化 Internationalization

「複数の国や地域の文化、言語および習慣を扱うためにコンピュータ・システム/ソフトウェアを汎用化するプロセス」(共立出版「国際化プログラミング」第1章より)

– Internationalization I18N (アイエイティーンエヌ)

I と n の間に18文字ある

– 地域化 Localization

二つの意味で使われる

- 日本語化, 中国語化, ドイツ語化などの各国語対応を行うこと
- I18N を実現した後、各国語対応を行うこと

L10N は後者 (I18N が前提)

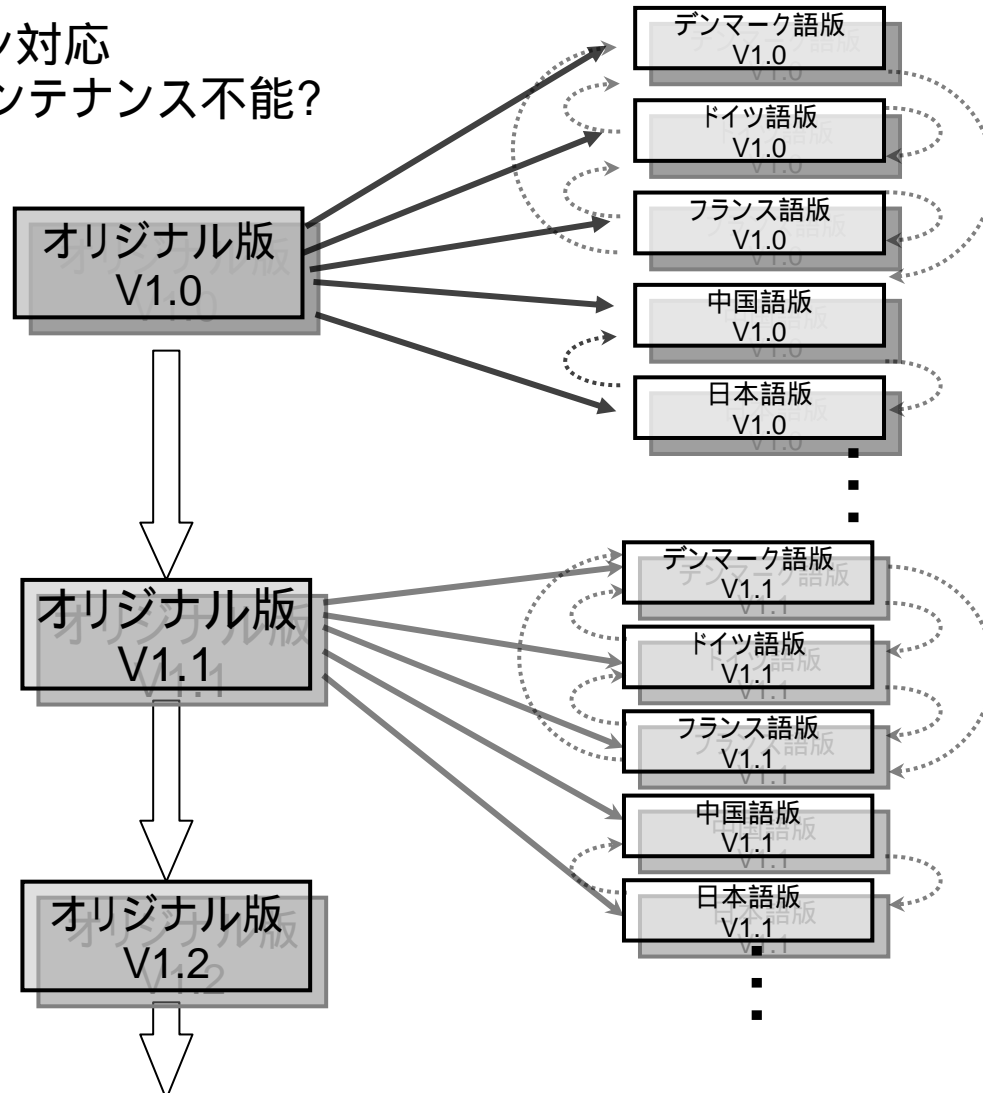
国際化JPは、I18N Company

国際化を考慮しない開発

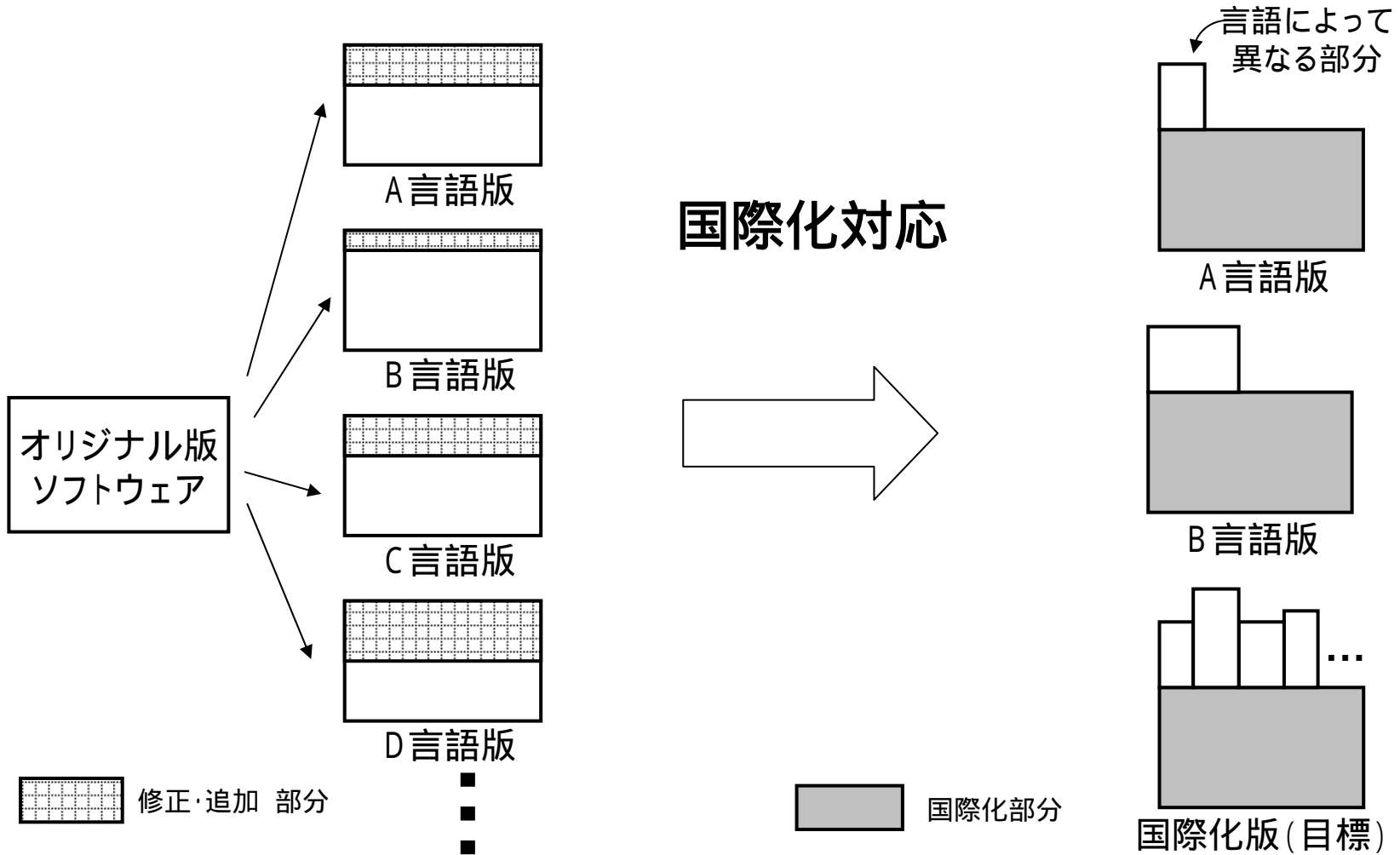
複数のバージョン対応
ここまで来るとメンテナンス不能?

ローカライゼーションは
時間と手間がかかる。

うまく管理しないと、
信じられないくらい大きな
無駄が発生する。



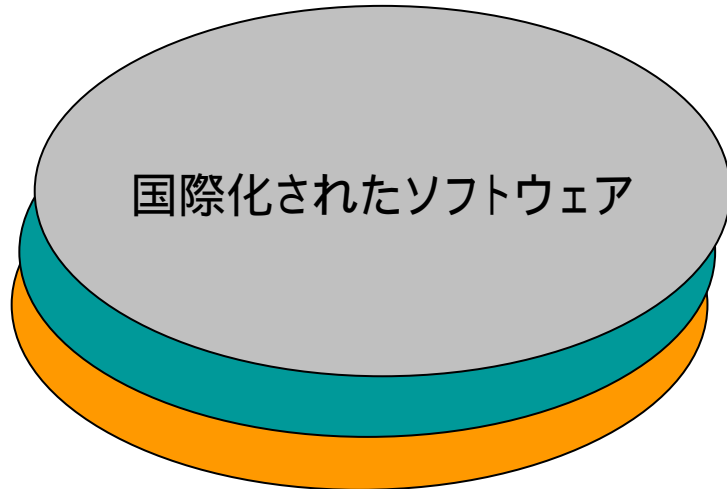
ソフトウェアの国際化とは



国際化されたソフトウェア

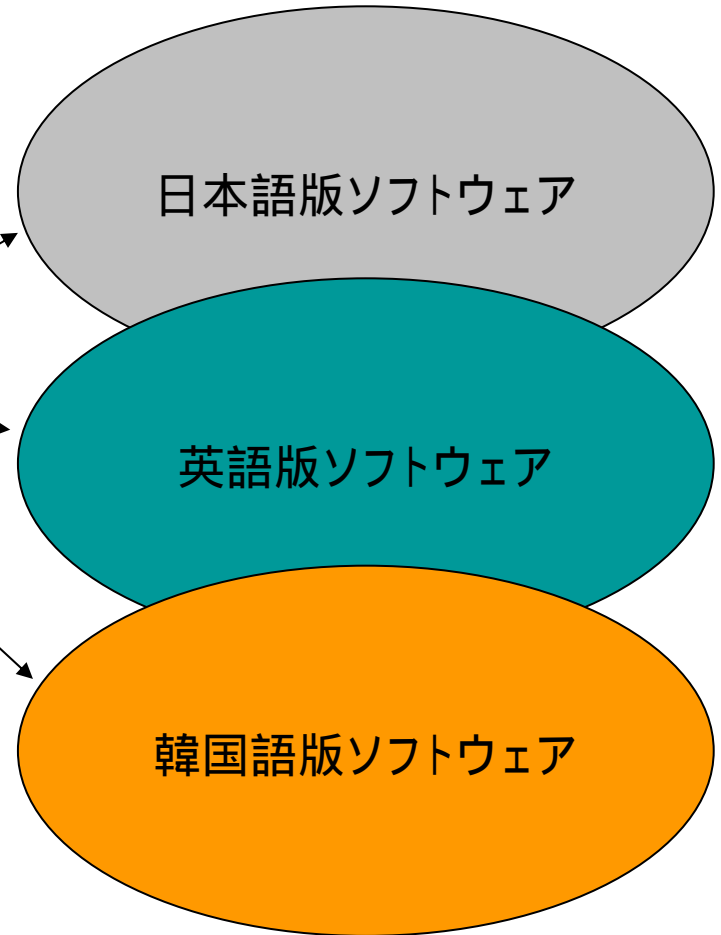
環境ごとに動作が自動で、あるいは
設定で切り替わる

同じソフトウェアで複数の言語を使い
分けることができる。



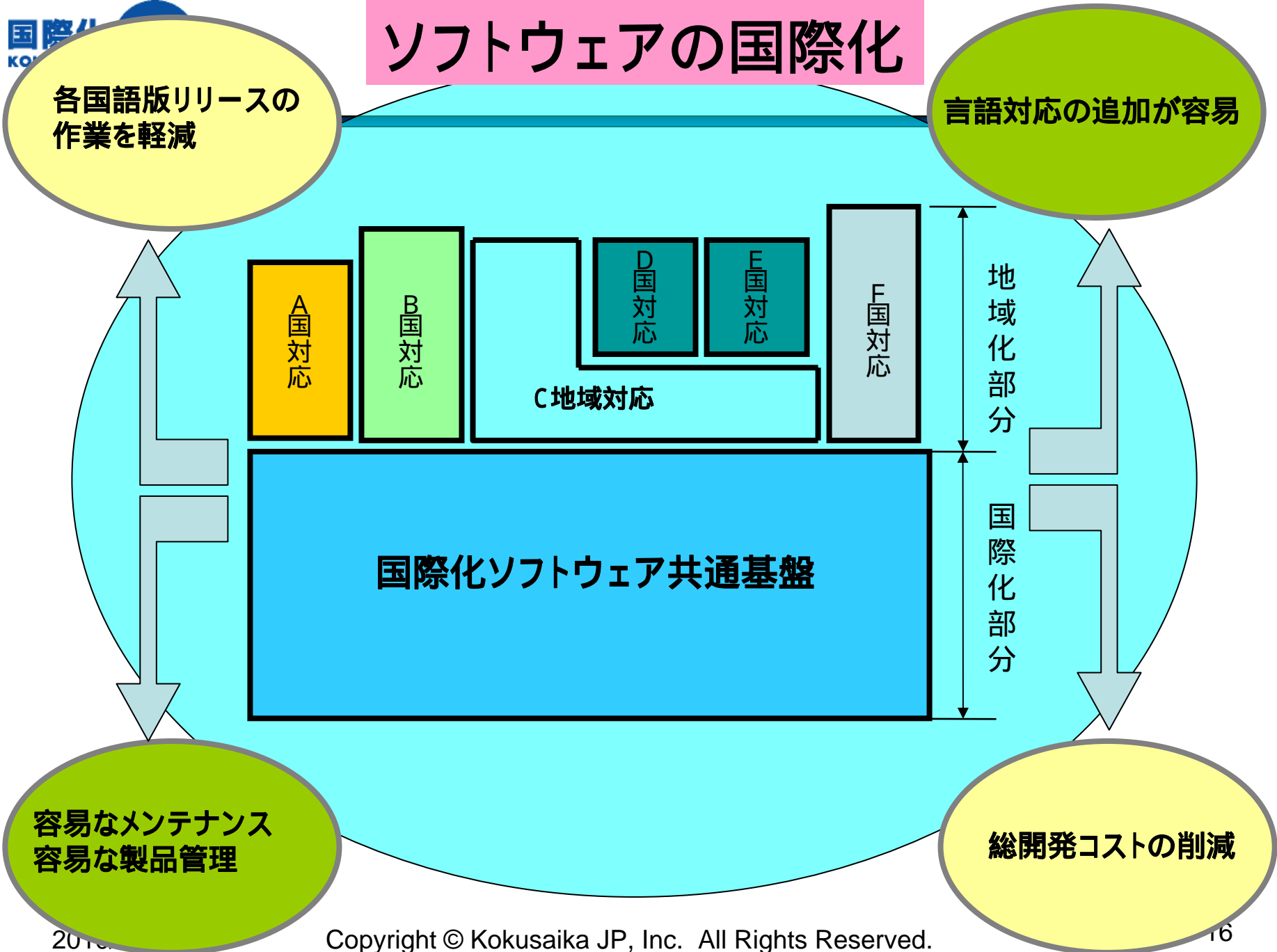
シングルソース、シングルバイナリの実現

OSの環境で切り替わるのが一般的だが、
アプリが独自に管理することもある。

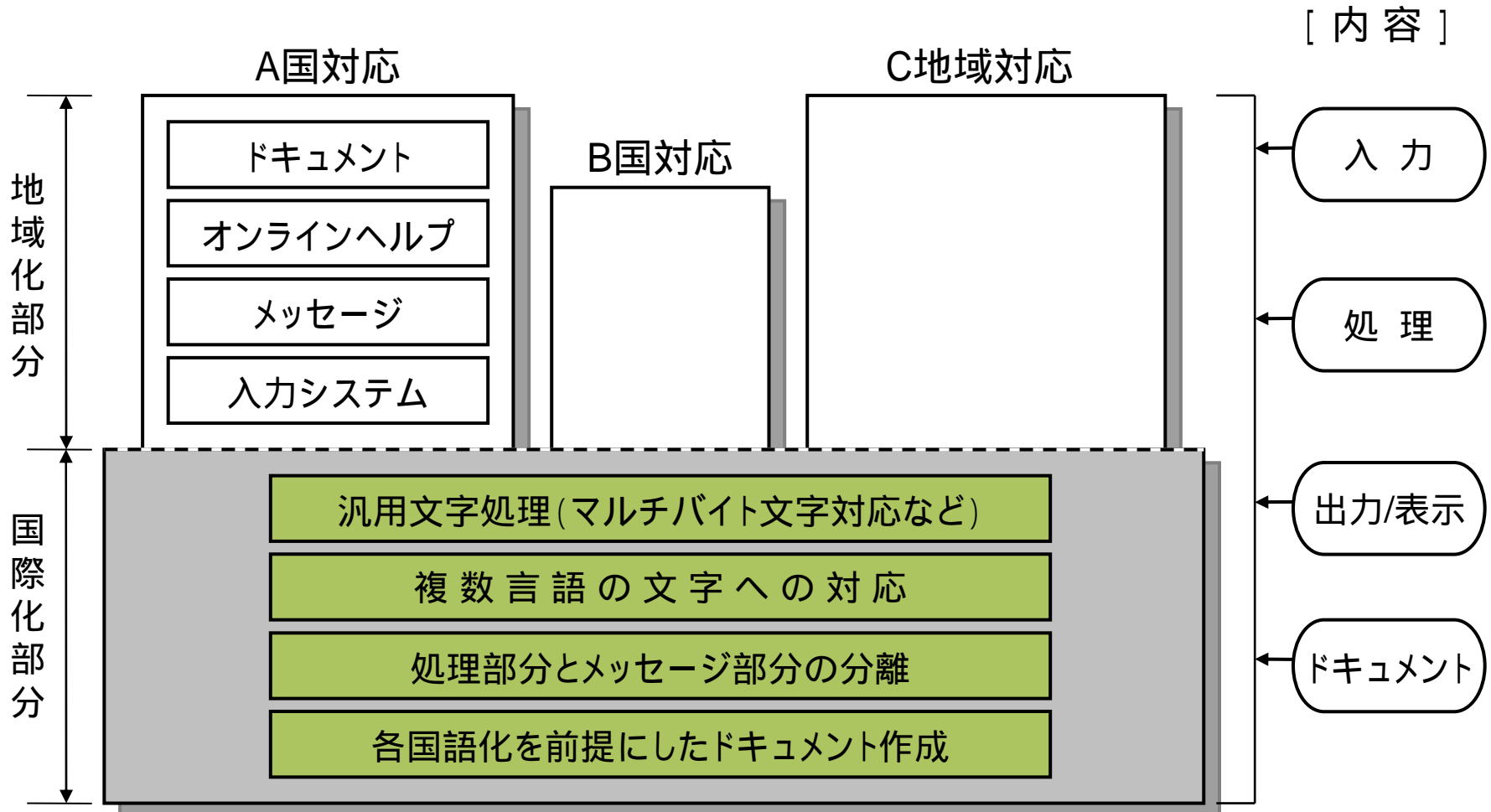


あたかも、それぞれの言語・地域のため
に作られたような機能および動作

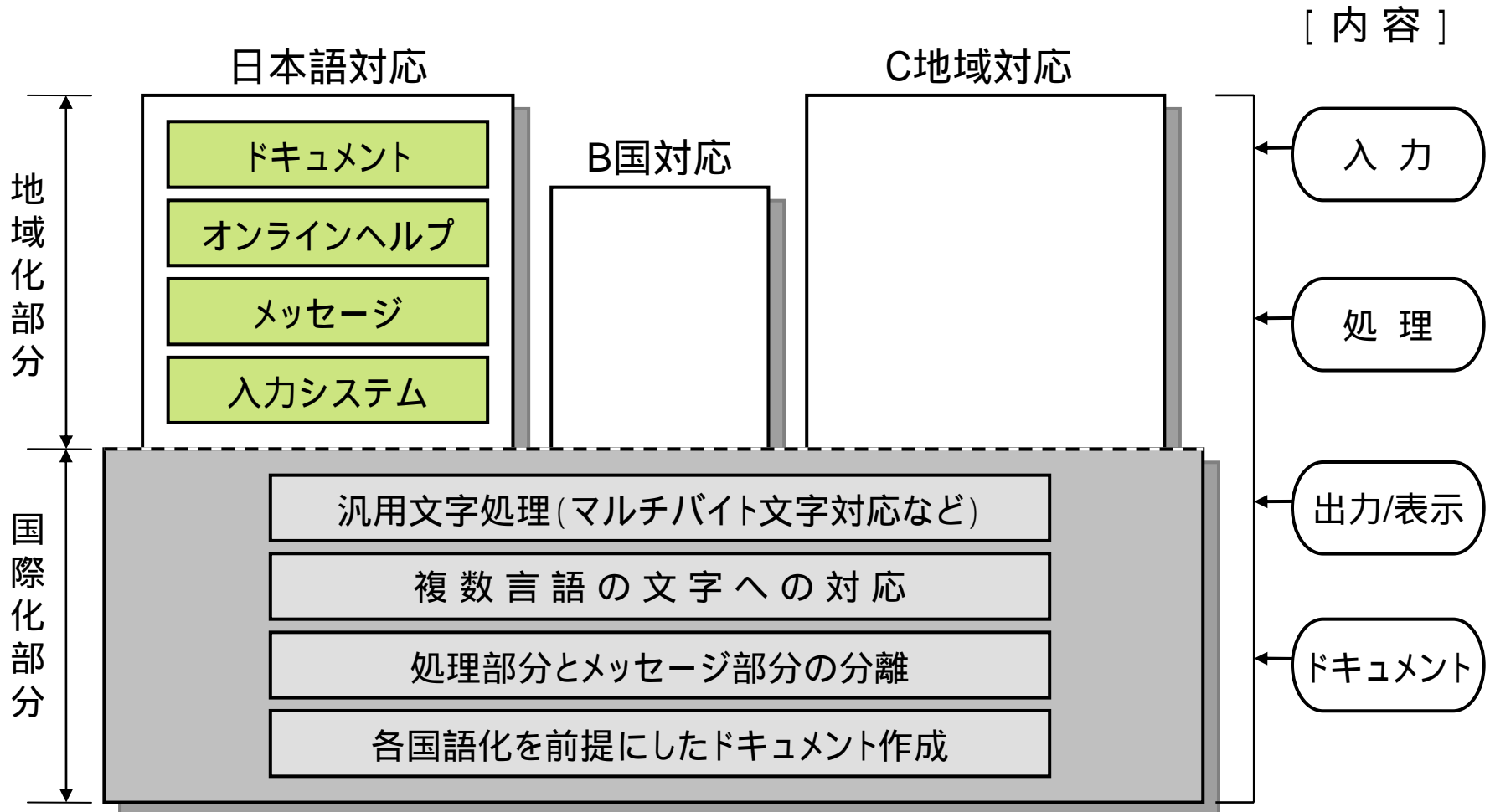
ソフトウェアの国際化



ソフトウェアの国際化とは



ソフトウェアの日本語化とは



3. 国際化プログラミング



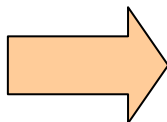
ソフトウェア国際化の技術

- どうやって国際化を実現するか？
到達点: **シングルソース & シングルバイナリ**の実現
- **シングルソース & シングルバイナリ**を実現するための基礎技術
 - 環境を切り替える仕組み(ロケールなど)
 - 文字や環境依存のデータを汎用的に処理する仕組み(データ型、言語や文化に依存しないAPI)
 - 標準化が重要(ほぼ実現できている。これができないとなんでも自分で抱え持たないといけないことになる。)
 - 依然として、開発者の意識に依存(ここをなんとかしないといけない)

メッセージ外部化処理の概要

埋め込みメッセージを自動で外部化

```
main()
{
...
printf("住所を入力してください")
...
printf("ありがとうございました")
...
}
```



```
main()
{
...
printf(
MSGGET(FIEL1, "MSG1"))
...
printf(
MSGGET(FILE1, "MSG2"))
...
}
```

外部ファイルからメッセージを取得するようにソースコードを変更

FILE1:

MSG1 "住所を入力してください"
MSG2 "ありがとうございました"

- ソースコード中に埋め込まれたメッセージテキストを外部ファイルに書き出す。
- それと同時に、外部ファイルから読み込んでメッセージを表示するようにソースコードを書き換える。
- FILE1を翻訳して適切な場所に置くと、多言語アプリができる。

注) ソースコードの編集例は、概念を示すためのもので World Wide Navi を使用して実際に変換するものとは異なります。

ソフトウェア国際化の技術

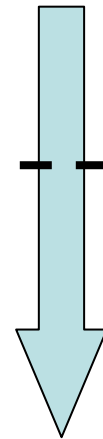
動作の仕組み

国際化API

どの言語でも呼び出すAPIは同じ
プログラマは、実行環境を意識する
必要がない

〔言語の場合の実装例〕
setlocale関数が呼び出された時点で、環境変数の設定に従って、国際化APIの参照先が変更される

複数のAPIを用意して切り替える方法やデータだけを切り替える方法など、実装方法はいろいろある。



下のレイヤーが**がんばっている**
環境に応じた適切な動作をするように作られている

特定言語環境用処理を切り替え

切替の例：
日付の表現方法
文字種の判定
1文字の取得

4. 文字コード Unicode 以前



- 1文字を表現する長さの変化
 - 7bit (128文字までの表現能力)
ASCII
 - 8bit (256文字までの表現能力)
ISO Latin 文字、JIS Roman + 半角カタカナ
 - ダブルバイト(16bit、単位がbitからbyteへ1 2)
 - マルチバイト(1,2,3,4)

 - 64 x 64 の平面にマップする
 - 平面を組み合わせる、切り替える仕組み

移りゆく文字処理のコード

- 文字処理

- 8ビットクリーン(7 8)
新しい文字集合をこれまでの器に入れて使う。MSB(Most Significant Bit 最上位ビット)が1でも処理が通るようにプログラムを書き換える。
- マルチバイト対応
新しい文字集合をこれまでの器に入れて使うか(MSBが1かどうかで処理を切り分けることが行われていた。)、器の大きさを変えてプログラムを作り直す。
- 文字コード体系の構築
問題が起こりにくい、処理しやすい仕組みを入れる シフトJIS、EUC
日本では複数文字コードが混在するため、誰もが文字コードの勉強をする必要があった。

- プロセスコードという概念

- 処理する文字コードのサイズを実装が決める。C言語のwchar_t など。
- バイト数、文字数、カラム数に左右されない文字を処理するための美しい世界

5 . Unicode の出現 黒船？救世主？



統一することに意味がある

- Unicode といえば 16ビット(だった)
- 海外の情報、国内の情報いろいろ
- 『統一することに意味がある』
 - 多くの人がそう考えた
 - 待望の統一コード誕生。世界に光がふりそそぐ？



Unicode で共通の統一された処理を！

- プロセスコードにユニバーサルコードセットを使う、
けがれなき美しい世界
- ファイルコードに使うための工夫 UTF-8

6. 美しい世界を築くユニバーサル コードセットの幻想



- 16ビット幻想の崩壊
 - JIS規格の文字でも64ビットで表現されるものも
- サロゲート文字、異体字セレクタ
 - 文字コードが機能を持ちはじめた？
 - フラットテキストの世界から離れていく
- 誰ががんばればいいのか？
 - たぶん、ユーザーではない
 - 開発者も全員じゃない
- 文字コードはITの基盤。本来安定性が求められる。
- うまくコントロールしていく必要がある。それは世界平和や安心して住める地球環境を求めて国や民族を越えて仕組み作りをしていくのと同じかもしれない。

日本から世界へ、世界から日本へ
よいソフトウェアを世界中の人に届けよう

国際化JP株式会社

since June 2006

<http://www.kokusaika.jp/>



写真： 高度8千メートルの機内から
世界の最高峰ヒマラヤ山脈を撮影。
(2009年2月)