

## 共同研究報告

## M. 多言語プログラミング環境の研究

## 研究者

代表者 野口 健一郎 神奈川大学理学部情報科学科 教授  
 共同研究者 今城 哲二 日立製作所ソフトウェア事業部

## 研究の概要

情報処理の多言語化 (Multi-lingualization) は、文字コードの国際標準化から始まり、インターネットのウェブ・ページの多言語化で大きく前進した。しかし、これまでの多言語化は文字や文書の範囲に留まっていると言ってよい。我々は、情報処理の根幹をなすプログラミングにおける多言語化の研究を進めてきた。

本研究では、昨年度に引き続き、Java 言語を共通の媒介言語として、ある国または地域の自然語をベースとしたプログラム記述 (自国語プログラムと呼ぶ) を、Java プログラムへと変換すること、またその逆変換も可能とすることの研究を行った。

本年度は、Java プログラムで多用される JavaAPI (Java の持つ標準クラス、API = Application Programming Interface) を対象に、JavaAPI で用いられる変数名やメソッド名の自国語表記を定め、それを英語ベースの JavaAPI との間で相互変換する方法を研究した。この相互変換のために、これまでの変換表に代わって、変換辞書を用いる方式とした。辞書に、API を構成する要素の単語を登録しておき、この辞書を用いて API 用語を自動変換するようにした。この変換は、自然語の翻訳に近いものになる。

この方式を実装した結果、かなり広い範囲の Java プログラムを自国語に変換でき、またその逆変換も可能になった。これにより、Java ベースの自国語プログラミングの実用性が従来に比べて飛躍的に向上した。

## 研究の成果

## 1. 実現方式

## (1) 方式概要

本年度実現した方式の概要を図1に示す。

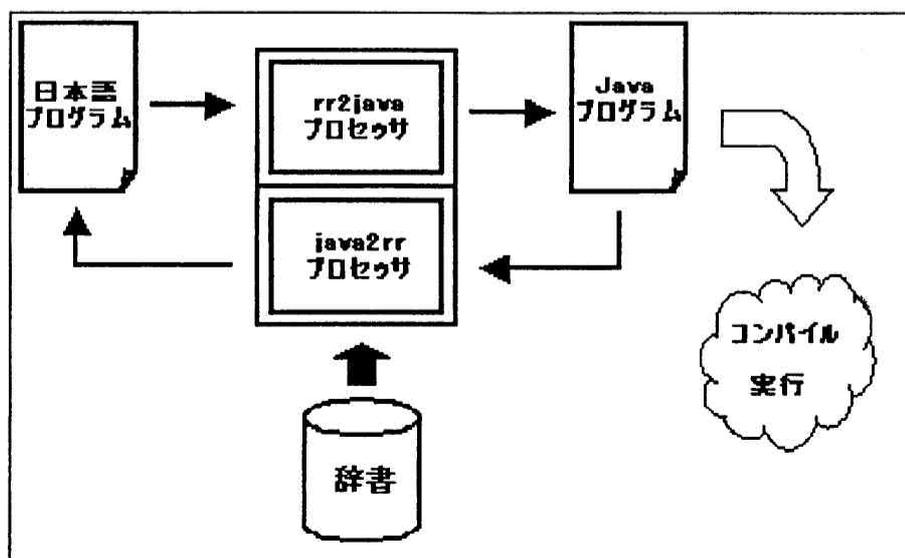


図1 変換辞書を用いた変換方式

変換プログラムは辞書を参照することにより、次を行う。

- ・ 自国語表記プログラムを Java プログラムへ自動変換
- ・ Java プログラムを自国語表記プログラムへ自動変換

## (2) 辞書の構成

辞書に登録する対象は次のものである。

- ・ プログラミング言語のキーワード
- ・ プログラミング言語の API を構成する単語

これらのものにつき、次の情報を辞書に含めた。

- ・ Java 表記
- ・ 自国語表記
- ・ 品詞情報

辞書の一部を図 2 に示す。

Java	日本語	品詞
class	クラス	キーワード
public	公の	キーワード
if	もし	キーワード
int	整数	キーワード
new	新規の	キーワード
draw	描く	動詞
String	文字列	名詞
set	設定	動詞
Color	色	名詞
get	取得	動詞
Image	画像	名詞

Java	日本語	品詞
next	次の	形容詞
Token	トークン	名詞
current	現在の	形容詞
Thread	スレッド	名詞
mouse	マウス	名詞
Clicked	がクリックされた	過去分詞
is	?	be動詞
Alive	生存	形容詞
push	押す	動詞
Back	後に	副詞
IOException	入出力例外	名詞

図 2 変換辞書の内容 (一部)

## (3) 変換アルゴリズム

JavaAPI で用いられている名前 (の殆ど) は次のような構文構造になっている。

- 単語一つ
- 単語二つの並び
- 単語 + 前置詞 + 単語

これに対応させて、日本語の場合の表記は次のようにした。

- 単語一つ
- 単語二つの並び

ただし、英語ベースの JavaAPI とは語順が逆になることもある。

- 単語 + 助詞 + 単語

このような構文構造を持った API 用語の変換には、各単語の品詞情報が必要である。それを利用して、自然語の機械翻訳と同様に、次の手順で変換を行った。

- ① 名前の構文解析：単語への分解

- ②単語毎の変換
- ③品詞情報に基づく名前の合成

## 2. 結果

辞書に、通常の Java の参考書で用いられているレベルの JavaAPI に含まれる単語約 500 個を登録した。

教材に用いている多くの Java プログラムを、

- ・日本語表記へ変換
- ・日本語表記のものを Java に逆変換

することができた。

Java プログラムから自動変換した日本語表記プログラムの例を図 3 に示す。

## 3. 今後の課題

### (1) 辞書の拡充

「コア JavaAPI」の用語を全て辞書化したい。

### (2) 変換アルゴリズムの拡張

3 単語以上の場合の変換アルゴリズムなど。

### (3) API 変換の localization

キーワードの場合と違って、API 用語の変換は難易度はそれほど高くないものの、一種の機械翻訳である。したがって、変換する対象言語のペア毎に、別々の処理が必要になる。本年度の実装は Java (すなわち英語) と日本語の間の変換に特化した作りになっている。API 用語変換の部分で、対象言語ペア毎に複数の実装を組み込めるようにし、実行時に該当する変換処理を呼び出せるような構造に拡張すべきである。

### (4) プログラムの表示形式 (レンダリング) に関する拡張

初年度の研究では、キーワードは太字で、また API 用語は斜字で表現することも可能にした。本年度の実装ではこの部分はやり残している。

さらに、色を利用したレンダリングも検討したい。

実現技術としては XHTML または XML を利用するのがよいだろう。

### (5) 記述実験および読解実験

実際に記述および読解の実験を行い、実際の効果の確認および方式改善へのフィードバックを行いたい。

## 参考文献

- [1] 鈴木康彦, 野口健一郎, 後藤英一: Java をターゲットにした自国語プログラミングの実験, 情報処理学会 第 58 回全国大会、5M-08、1999 年 3 月
- [2] James Gosling, Bill Joy, Guy Steele, Gilad Bracha: The Java™ Language Specification Second Edition, Addison-Wesley (2000)
- [3] Java™ 2 SDK, Standard Edition, Version 1.3 ドキュメント  
<http://java.sun.com/j2se/1.3/ja/docs/ja/>
- [4] Java™ プラットフォーム 1.3 API 仕様  
<http://java.sun.com/j2se/1.3/ja/docs/ja/api/>
- [5] The Unicode Consortium, The Unicode Standard, Version 2.0, Addison-Wesley Developers Press (1996)
- [6] 松下尊信: 自国語プログラミングにおける自動変換範囲の拡大、2001 年度卒業論文 (野口研究室) (2002)

```

import java.applet.*;
import java.awt.*;
import java.awt.Image;

公の クラス Animation は アプレット を拡張 実行可能 を実装 {
    画像 がぞう[] = 新規の 画像[10];
    揮発性の スレッド 描画屋 = 空;
    整数 n = 0, x = 50, y = 50, 画像幅, 画像高;
    色 いろ;

    公の 空の 初期化0 {
        繰り返し (整数 i = 0; i < 10; i++)
            がぞう[i] = 画像を取得(ドキュメントベースを取得0, "images/" + i + ".gif");
        いろ = 新規の 色(200, 170, 80);
        背景を設定(いろ);
    }

    公の 空の 開始0 {
        もし (描画屋 == 空) {
            描画屋 = 新規の スレッド(自分);
            描画屋.開始0;
        }
    }

    公の 空の 実行0 {
        スレッド この糸 = スレッド.現在のスレッド0;
        ループ(この糸 == 描画屋) {
            再描画0;
            トライ {
                スレッド.眠る(200);
            } 捕まえる(割り込み例外 例外) {
            }
            n = (++n)%10;
        }
    }
}

```

```

公の 空の 描画(グラフィックス 図) {
    画像幅 = がぞう[n].幅を取得(自分);
    画像高 = がぞう[n].高さを取得(自分);
    図.画像を描く(がぞう[n], x, y, 自分);
}

公の 空の 更新(グラフィックス 図) {
    図.色を設定(いろ);
    図.四角形を塗りつぶす(x, y, 画像幅, 画像高);
    描画(図);
}

公の 空の 停止0 {
    描画屋 = 空; // Thread の stop()は推奨されない。
                // Thread は自分で終わらせるべし。
}
}

```

図3 日本語表記プログラムの例 (Java プログラムから自動変換したもの)