

共同研究報告

P 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

情報処理の根幹をなすプログラミングにおける多言語化 (Multi-lingualization) の研究を行っている。これまで、Java 言語を共通の媒介言語として、Java プログラムを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) へと変換すること、またその逆変換も可能とすることにつき、基本方式を確立した。[1], [2]

本年度は、自国語プログラミングをより実用に近付けるために、日本語ベースの自国語プログラミング用の変換辞書の拡充とそれを利用した変換方式の改良の研究を行った。また、プログラムの読みやすさを増すためのプログラミングテキストの表示方式 (レンダリング) の研究を行った。研究の結果得られた変換辞書および変換ツールを利用して、実際の Java プログラムを日本人にとって読みやすい自国語 (日本語) プログラムに自動変換できることを確かめた。

研究の成果

1. 全体の方式

図1に示す。

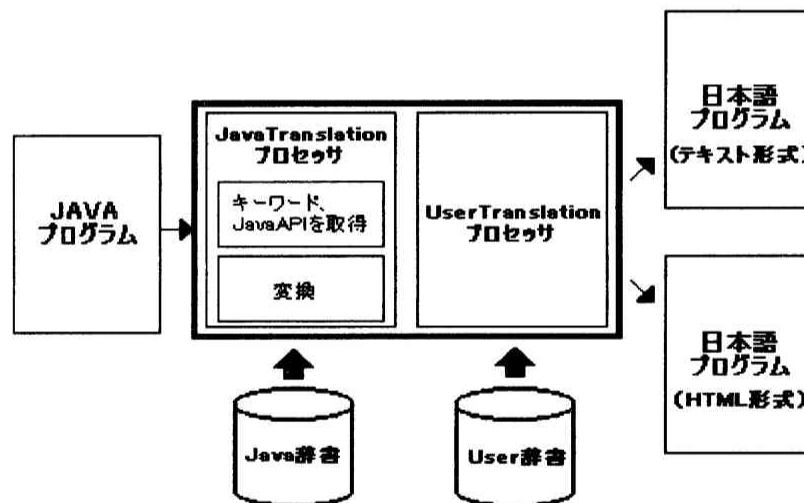


図1 システムの構成図

2. 変換辞書の拡充と変換方式の改良

Java プログラムで自国語表記したい用語 (名称) には次の種類がある。

①キーワード

②JavaAPI (Java の持つ標準クラス、API = Application Programming Interface) で用いられる変数名やメソッド名

③利用者定義の変数名やメソッド名

これらの用語の変換に際しては、②の用語が非常に数が多いことへの対処の仕方、また②と③の用語をどう区別すべきか、という課題がある。

●JavaAPI 用語の辞書登録法

JavaAPI 用語は複数の英単語の組み合わせで成り立っている。まず、JavaAPI 用語で使用さ

れている単語を、変換辞書に登録する作業を行った。主要なパッケージを対象に、約1200語を登録した。

次に、複数用語の組み合わせで成り立つJavaAPI用語を、機械翻訳の手法を用いて、日本語用語へ自動変換する方式につき、昨年度の研究で開発した基本の方式を改良して、自然な日本語表現で、かつ首尾一貫した用語が得られるようにした。JavaAPI用語約3000語について、自然な日本語用語が得られることを確認した。

●JavaAPI用語と利用者定義用語の区別

JavaAPI用語の自動変換に伴い、利用者定義用語まで一緒に変換されてしまう、という副作用があった。これは、プログラム中のimport文を利用して、JavaAPI用語かどうか判定することにより解決した。さらに、利用者定義用語の変換用に別の辞書を設けた。

3. プログラミングテキストの表示方式（レンダリング）

変換ツールの出力形式として、次の2形式を選べるようにした。

①テキスト形式

②HTML形式

HTML形式の場合に、1.(1)に挙げた3種類の用語ごとに、表示の色や字体（太字やイタリックなど）を変えるようにした。これにより、読みやすいテキストが実現された。

4. 実プログラムへの適用

変換ツールはJavaプログラムとして作成したが、そのプログラム自身を自国語表記に変換する実験を行い、問題なく自国語表記プログラムが得られることを確認した。例を図2に示す。

5. 今後の課題

(1) 日本語プログラムからJavaプログラムへの変換

今年度の変換方式の改良は、Javaプログラムから日本語プログラムへの変換だけを実現した。さらに、逆方向の変換も可能にすることはさほど難しくはない。このためには、変換辞書を2レベルに分け、単語を登録した辞書は、JavaAPI用語の対応表を作成するためだけに使い、出来上がった対応表を第2の辞書として、プログラムの変換に用いるようにすればよい。この改良により、双方向の変換も可能になるとと思われる。

(2) 他言語への対応

JavaAPI用語の変換が一種の機械翻訳となることから、今年度のものは日本語に特化した変換ツールになってしまっている。(1)で述べたように、JavaAPI用語の対応表作成を独立フェーズにすることにより、言語依存の部分はそこに集約できる可能性がある。

(3) 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系統的に行い、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] 松下尊信, 野口健一郎: Java をターゲットにした自国語プログラミングにおける辞書を用いた自動変換, FIT (情報科学技術フォーラム) 2002, B-7, 2002年9月

[3] 滝本修平: 自国語プログラミングの実験—JavaAPIの変換の改良—, 2002年度卒業論文 (野口研究室), 2003年2月

```

import java.lang.reflect.*;
import java.util.ArrayList;
import java.io.*;

/**
 * キーワード、JavaAPI 専用翻訳機械
 **/
公開 クラス 翻訳機械2 拡張 翻訳機械 {

    類別 類別 = 新規の 類別();

    非公開 メソッド[] メソッド群 = ナル;
    非公開 フィールド[] フィールド群 = ナル;
    非公開 コンストラクタ[] コンストラクタ群 = ナル;
    非公開 クラス[] インターフェイス群 = ナル;

    非公開 配列リスト API リスト = 新規の 配列リスト();

    公開 翻訳機械2() {}

    公開 文字列 API 操作(文字列 英語) 投げる 10 例外 {
        Lang();
        (整数 i = 0; i < API リスト. サイズ(); i++) {
            // キーワード及び JavaAPI なら変換
            もし(API リスト. 得る(i). 等しい(英語)) {
                戻す 操作(英語);
            }
        }
        戻す 英語;
    }

    /**
     * JavaAPI を取得
     **/
    公開 空の JavaAPI を読む(文字列 クラス) {

        類別. クラスを設定する(クラス);

        メソッド群 = 類別. 全てのメソッド群を取得する();
        (整数 i = 0; i < メソッド群. 長さ; i++) {
            API リスト. 追加する(メソッド群[i]. 名前を得る());
        }
        フィールド群 = 類別. 全てのフィールド群を取得する();
        (整数 i = 0; i < フィールド群. 長さ; i++) {
            API リスト. 追加する(フィールド群[i]. 名前を得る());
        }

        コンストラクタ群 = 類別. 全てのコンストラクタ群を取得する();
        (整数 i = 0; i < コンストラクタ群. 長さ; i++) {
            API リスト. 追加する(類別. API を取得する(コンストラクタ群[i]. 名前を得る()));
        }

        インターフェイス群 = 類別. 全てのインターフェイス群を取得する();
        (整数 i = 0; i < インターフェイス群. 長さ; i++) {
            API リスト. 追加する(類別. API を取得する(インターフェイス群[i]. 名前を得る()));
        }
    }

    /**
     * パッケージ java.lang に含まれる JavaAPI を取得
     **/
    非公開 空の Lang() 投げる 10 例外 {
        ファイル ファイル = 新規の ファイル("apiLang.txt");
        バッファされたリーダー リーダー = 新規の バッファされたリーダー(新規の ファイルリーダー(ファイル));
        文字列 行;
        ループ((行 = リーダー. 行を読む()) != ナル) {
            API リスト. 追加する(行);
        }
    }
}

```

図 2 自動変換したプログラムの一部 (HTML 形式)