

プログラミングを題材とした国際交流授業の提案*

兼宗 進^{*}, 李 元揆[†], 久野 靖[§]

一橋大学 総合情報処理センター^{*} 高麗大学[†] 筑波大学大学院 ビジネス科学研究科[§]

〒 186-8601 東京都国立市中 2-1^{*}

kanemune@cc.hit-u.ac.jp^{*} lee@comedu.korea.ac.kr[†] kuno@gssm.otsuka.tsukuba.ac.jp[§]

概要

プログラミングを中心的なコミュニケーション主題とする国際交流授業のモデルを提案する。プログラミング言語は人工言語であり、変数名、命令語などを各国の言語で記述するようにできる一方、自然言語と比べて構文を共通にできることから、単語レベルの翻訳で互いの意図を理解しやすいという利点を持つ。本発表では、多言語で記述可能な教育用プログラミング言語を用いて、異なる言語を話す地域との交流を行うための学習環境を提案する。

1 はじめに

従来の、電子掲示板や電子メールを中心とした国際交流授業では、共通の興味を持続しにくい、共通の背景を前提にできないので意志の疎通が難しい、言葉の理解が壁になる、といった問題が存在した。

今回提案するモデルでは、プログラミング言語を用いて作品を作成し、それをサーバーを介して交換する。プログラミング言語ドリトルではプログラムを各国語で記述できるため、自国語を用いて抵抗なく作品を作ることができ、さらにそれを単語単位で相手側の言語の語彙に置き換えることで相手側にとっても抵抗なく読解することができる。

2 プログラミングを題材とした国際交流

初中等教育で行われる情報教育の中で、プログラミングの体験はIT技術の根幹に触れることのできる機会として重要な役割を持っている。

プログラミングを体験するためには、適切な学習用のプログラミング言語が不可欠である。筆者らは過去にドリトルを提案し、次の成果を示してきた。

(1) 習得が容易な言語を用いることで、「言語そのものの習得」が目的にならず、「その言語を用いて作品を作ること」を目的にできること。

(2) GUIプログラムなどある程度実用的なアプリケーションプログラムを容易に作ることで、生徒たちが強い興味を持って学習に取り組み、計算機が動作する原理について体験的に学習できること。

プログラミング言語は人工言語であり、国が違ってても言語の構文は変わらないため、自然言語のような翻訳の難しさはない。そこで今回提案するツールでは、プログラムのうち、予約語に相当する語を自国語に変換して表示する機能を用意した。

C言語など通常の言語では、プログラムに国による違いは存在しないが、初中等教育には敷居が高いという問題が存在する。今回採用したドリトルは、命令語などの名前に各国語を使うため、小学校から高校までの生徒にも抵抗なく受け入れられる言語である [6]。その代わりに、国ごとに言語が異なることになるが、これについては単語単位の置換により、容易にソースコードを他の言語に変換することができる。

International Collaboration in School Based on Multilingual Programming Language, Susumu Kanemune^{}, WonGyu LEE[†], Yasushi Kuno[§] (Hitotsubashi University^{*}, Korea University[†], Tsukuba University[§])

提案モデルは初中等教育のほか、大学などの高等教育でも利用可能である。現在は、韓国の大学との間で予備的な実験を進めており、今後は K12 の環境での評価を計画している。

3 ドリトルの国際機能

3.1 プログラミング言語ドリトル

ドリトル [1][5][7] はオブジェクト指向でありながら初心者または初中等教育における若年層から使うことのできるプログラミング言語である。

ドリトルは日本語で記述できることを特徴として設計された。元々は日本国内で使うことを想定しており、英語を習っていない小学生、または英語に熟練していない中学生が自国語で書けることを目標としたものである。図 1 に、ドリトルの画面と実行例を示す。



図 1: ドリトルの画面と実行例

3.2 国際版ドリトル

その後、海外での需要が出現したため、英語版と韓国語版を開発した。ドリトルには予約語が存

在しないため、標準の識別子を各国語の単語に置き換えることで複数の言語に対応することができた。具体的には、ドリトルの自動実行ファイル (dolittle.ini) の中で、英語と韓国語に対応する日本語の命令語を実行するメソッドを定義している。図 2 に、外国語命令の定義例を示す。この例では x を受け取り、それを引数にして「歩く」を呼び出す forward というメソッドを定義している。

```
タートル:forward= 「|x|!(x) 歩く」
```

図 2: 外国語命令の定義例

図 3 に、韓国語版の画面を示す。

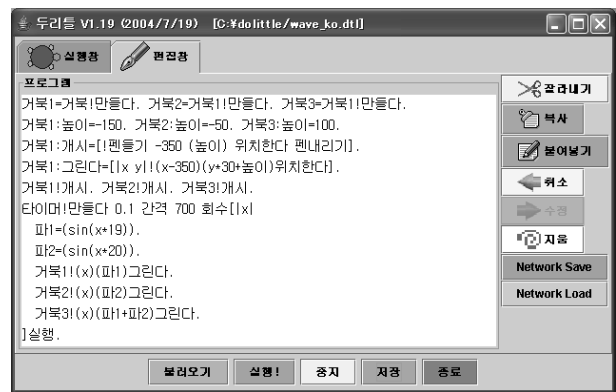


図 3: 韓国語版の画面

3.3 交流授業でのソースコード交換

ドリトルに限らず、プログラムを題材とした国際交流授業を行う場合、学校間でのソースコードの交換には次のような手段を用いるのが一般的であった。

(1) プログラムをメールに添付して送る。メールでの添付は、送る側はプログラムファイルをドリトルのインストールディレクトリ (通常は C:\dolittle) から探してメールソフトに添付する。そして、受ける側は添付されたファイルをドリトルのインストールディレクトリに保存する必要がある。これは手順が多く、生徒が作業することは容易ではない。

(2) プログラムを HTML に記述して WWW で公開する。プログラムが改行などを含めて正しく表示されるように HTML を記述する必要がある。その他、サーバーにアップロードする手順などがあり、自作の作品プログラムを広く公開する手段

としては効果的だが、生徒が作業することは容易ではない。

(3) プログラムを電子掲示板 (BBS) に書いて公開する。プログラムをドリトルの画面と Web ブラウザの間でカット&ペーストする。メール添付ほど難しい作業ではないが、1画面に収まらない部分を含めてプログラム全体を切り貼りする必要がある点を考えると、生徒が正確に作業することは容易ではない。

以上は同じ言語 (国語) を使用する生徒同士のファイル交換であったが、異なる言語を使用する生徒同士のファイル交換を考えると、さらに次のような問題が生じる。

(a) 文字コードの問題。ドリトルでは、ディスクに保存する際の文字コードは、使用する環境 (ドリトルを実行しているオペレーティングシステム) の標準の文字コードを採用している。Windows 環境では、日本語は Shift-JIS、韓国語は KSC である。これらのコードには互換性がないため、たとえば KSC コードで記述された韓国語のプログラムファイルを日本語ドリトルで読むことはできない。(表示が化けるだけでなく、文字コードレベルで正しく読み込まれない)

(b) 言語の違いの問題。たとえ文字コードの問題を解決したとしても、変数名や命令に日本語が使われている場合、それを韓国の生徒が理解するのは困難である。

4 プログラム交換サーバ方式

4.1 フレームワークの提案

上で述べた既存の方法が持つ問題、および言語の違いの問題を解決するために、インターネット上にプログラム交換サーバを用意し、ここを経由してプログラムを交換する方式を考案した。教室で生徒や教師が使うドリトル処理系 (Java 2 で記述) はこのクライアントとしてこのサーバと連携して動作する。サーバを経由することで、既存の方式より柔軟かつ容易にコードが交換でき、またサーバ上で変換処理を行うことにより、言語の違いの問題にも対応できる。

図 4 に、システム構成を示す。プログラム交換サーバは Perl で記述された CGI プログラムである。今回は Linux 上で動作を確認した。クライ

アントはドリトルが動く環境である [4]。今回は WindowsXP (日本語版、韓国語版)、Linux で動作を確認した。

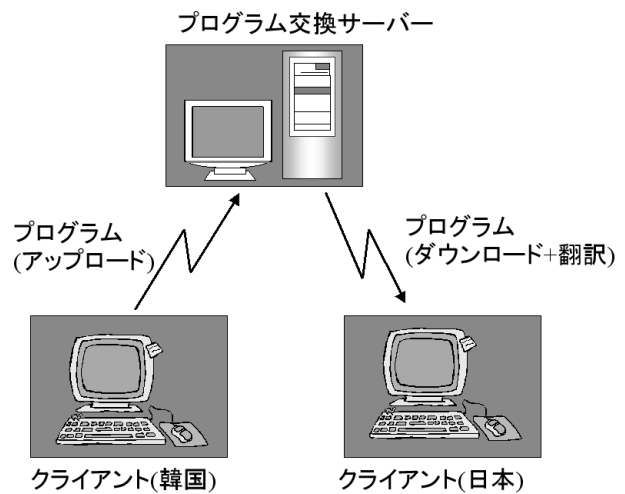


図 4: 翻訳サーバのシステム構成

4.2 サーバの機能

プログラム交換サーバは、ファイルの登録 (save) ファイルの一覧 (list) ファイルの読み出し (load) という 3 つの機能を提供する。

クライアントとの通信には http 上で XML-RPC を用いる。セキュリティのために、XML-RPC の接続時にはクライアントからドリトルであることの証明をキーとして送信し、認証を行っている。クライアントとの通信で使われる文字コードは UNICODE である。

ファイルの登録 (save) 処理では、クライアントからプログラムファイルを受け取り、サーバ上に保存する。今回の実装では、データベースを使わずファイルとして保存した。プログラムファイルはクライアントでは言語ごとの文字コード (Shift-JIS、KSC、ASCII など) で保存されるが、サーバ上には UNICODE で統一して保存される。

ファイルの一覧 (list) 処理では、サーバ上のプログラムファイルの名前を一覧としてクライアントに送る。

ファイルの読み出し (load) 処理では、クライアントから名前と言語名を受け取り、指定された名前のプログラムファイルを読み出す。サーバ上のプログラムファイルは、文字コードは UNICODE であるものの、内容は記述した言語のままである。

そこで、クライアントの言語(日本語、韓国語、英語など)に合わせて、プログラムを翻訳して送る機能を追加した。翻訳されるのは、ドリトルが解釈する識別子(命令語、プロトタイプ名)とユーザーが登録した識別子(変数名など)である。クライアント言語の判断には、ドリトルを実行するOS環境のロケール(LOCALE)が使われる。英語はen、日本語はja、韓国語はkoである。

表1に、ドリトルの識別子を翻訳するための対応表を、図5に翻訳処理の流れを示す。表1で、各カラムは言語ごとの識別子を表している。ある識別子に複数の表記が存在する場合には、その言語の2番目以降の表記を別名(alias)として記述する。

表 1: 翻訳のための対応表 (一部)

英語	日本語	韓国語	alias
collision	衝突	충돌	
click	動作	클릭	
root	ルート	루트	
turtle	タートル	거북	거북이
timer	タイマー	타이머	시계
button	ボタン	버튼	
field	フィールド	필드	텍스트필드 빈칸
list	リスト	리스트	목록

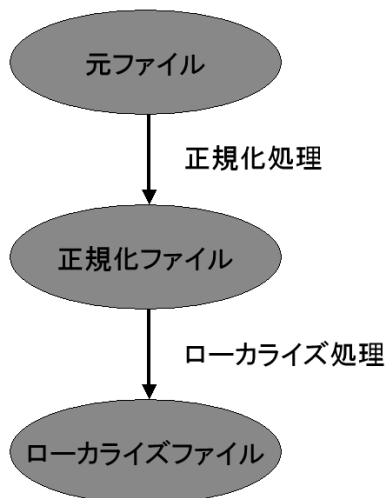


図 5: 翻訳処理の流れ

4.3 クライアントの機能

クライアントでは、編集画面に「サーバー保存」「サーバー読込」という2つのボタンを用意した。

サーバー保存ボタンを押すと、ファイル選択ダイアログが開き、ローカルファイルの一覧が表示される(図6)。ファイルを選択すると、そのファイルがファイル交換サーバーに送られる。



図 6: ファイル選択ダイアログ (アップロード)

サーバー読込ボタンを押すと、ファイル選択ダイアログが開き、サーバー上のファイルの一覧が表示される(図7)。ファイルを選択すると、サーバー上のファイルが編集画面に読み込まれる。



図 7: ファイル選択ダイアログ (ダウンロード)

このとき、サーバー側ではクライアントの動作言語に合わせた翻訳処理が行われるため、異なる言語で書かれたプログラムであっても、クライアントの言語に合わせた形でダウンロードされる。

5 国際交流授業での利用

ファイル交換サーバーを授業で使う際の課題を考察する。

5.1 ソースコード 以外の情報交換

授業で国際交流を行う場合には、互いにプログラム作品を交換するだけでなく、感想や改良点のコメントなどをフィードバックする手段が必要になる。

プログラムは生徒の作品を伝えることで共通の話題を提供するが、その後のコミュニケーションは自然言語で行われることになる。

双方向のやり取りを行うためには、静的なページではなく、動的に生成可能な環境を利用することが望ましい。一般的にはチャットや電子掲示板が使われるが、発言が流れてしまう(消えるまたは見えなくなる)という問題がある。国際交流での発言量はそれほど多くないことを考えると、Webブラウザからコメントを記述することが可能なWikiやBlogの活用が適していると考えられる。変更がRSSで公開されるため、授業を行う教員はWeb上の変更を随時知ることが可能である。

これらのツールは標準的なものをそのまま使うことも可能だが、授業で十分に活用するためには、次の改善を行う必要があると考えている。今後の課題として検討を進める予定である。

- メニューのローカライズ
画面のメニューやラベルがその国の言葉で表示されること。
- コンテンツの翻訳
生徒が書いた感想などが、翻訳されて表示されること。
- ドリトルとの連携
ドリトルの作品を画面内で実行できるとよい。

5.2 プログラム翻訳

プログラム交換サーバーにより、ドリトルを国際交流で使う際に問題となっていた異なる言語でのプログラム転送を可能にすることができた。

また、プログラム翻訳機能は、どの言語で書かれたプログラムをサーバーに登録しておいても、ダウンロード時にクライアントに適した言語に翻訳する。この機能によって、言葉の壁を超えて国

際的にプログラムを交換する環境の基礎を築くことができた。

しかし、いくつかの課題が残されている。大きな課題としては、翻訳できない語の扱いがある。現在の翻訳機能で翻訳される対象は、記号とドリトルが認識する識別子(命令語など)、あらかじめユーザーが登録しておいた識別子(変数名など)に限られているため、すべての識別子に対応しているわけではない。

今後は、次のような拡張を検討していく予定である。

- フリーの辞書を組み込む
辞書をデータベースに登録し、翻訳時に参照する方法である。
- オンラインのサービスを利用する
オンラインの辞書または翻訳サービスにhttpでアクセスし、機械的に翻訳を行う方法である。SqueakToys[2]のプログラムを日本語と英語で相互翻訳するTranSqueak[3]の試みがある。

Web上のサービスに対する機械的なアクセスは負荷や許諾の点で問題になる可能性があるが、フリーの辞書と合わせて効果が高いことが期待される。

6 おわりに

本稿では、初中等教育で活用可能なプログラミング言語ドリトルを国際交流に使用するアイデアを紹介し、プログラムが自動的に各国語に翻訳されるプログラム交換サーバーについて報告した。

現在は試作したサーバーを大学レベルで試している段階である。今後は、翻訳の向上とともに感想などを交換する学習環境を構築し、初中等教育での実践を進めて行きたいと考えている。

ドリトルの韓国語命令については、長野大学の和田勉教授と高麗大学の李研究室のメンバーに監修をお願いした。この場を借りて感謝いたします。

参考文献

- [1] Dolittle programming language. <http://kanemune.cc.hit-u.ac.jp/dolittle/>.
- [2] Alan Kay. Etoys and simstories in Squeak. <http://www.squeakland.org/author/etoys.html>.
- [3] Michael Ruger and Yoshiki Ohshima. Trans-squeak — making the world a smaller place On-the-fly translation of Etoy projects and instant messaging. In *The Second International Conference on Creating, Connecting and Collaborating through Computing*, pp. 90–96, 2004.
- [4] 兼宗進, 御手洗理英, 中谷多哉子, 福井眞吾, 久野靖. 学校教育用オブジェクト指向言語「ドリトル」の設計と実装. 情報処理学会論文誌, Vol. 42, No. SIG11(PRO12), pp. 78–90, 2001.
- [5] 兼宗進, 中谷多哉子, 御手洗理英, 福井眞吾, 久野靖. 初中等教育におけるオブジェクト指向プログラミングの実践と評価. 情報処理学会論文誌, Vol. 44, No. SIG13(PRO18), pp. 58–71, 2003.
- [6] 兼宗進, 中谷多哉子, 御手洗理英, 福井眞吾, 久野靖. 端末を飛び出したオブジェクト: 分散プログラミングを活用した情報教育の提案. 情報処理学会 情報教育シンポジウム (SSS2003), pp. 91–98, 2003.
- [7] 紅林秀治, 兼宗進, 岡田雅美, 佐藤和浩, 久野靖. 画面を飛び出したオブジェクト: 自立型ロボットを活用した情報教育の提案. 情報処理学会 情報教育シンポジウム (SSS2002), 2002.