

疑似体験によるJava学習システム

Learning System for Java by Doing of quasi

高津陽平
Yohei TAKATSU

伊藤小琴
Ogoto ITO Yoshitaka MAEKAWA

前川仁孝
Mitsuhiro IYODA

伊與田光宏

千葉工業大学
Chiba Institute of Technology

1. はじめに

近年、教育機関への情報教育導入と多様化により、授業や研究での利用を目的として、プログラミングを学習する学生が増加している。

プログラミング技術習得には学生の自習が求められる。しかし、構文理解が困難なことや、学習時間に対する達成感の低さから挫折し、苦手意識を持ってしまう学生も少なくない。また、授業を行う指導者は学生が構文をどの程度理解しているのかを把握するのが容易ではない。

そこで本研究では、視覚からの構文理解促進と、疑似体験による達成感を提供可能とさせるシステムの構築を目指す。

2. 学習形式の現状

学生の学習情報源には、指導者による授業の他に、書籍、CD-ROM、Webサイトがある。書籍、CD-ROMからの学習は情報が固定である。ゆえに学生は内容において、語彙の取り違えから、理解困難な状況が継続する。これは苦手意識を抱く要因となる。また、Webサイトでの他者の力による問題解決は、自分の学習進度を他者に伝達しなければ成立しない。

3. 学習方法の現状

プログラミングの学習方法は、個人の適性により、積み上げ型、解決型の2通りに大別される。

積み上げ型学習法は構文を学習後、プログラミングを行う方法である。この方法は、問題に対し、複数の処理法を考案する、柔軟なプログラミング能力が習得可能である。しかし、構文理解に多くの時間を費やすため、学生の動機付けを低下させる。

解決型学習法はプログラミングを行う過程で構文を学習する方法である。この方法では、学習初期からプログラミングを体験可能なため、学生が楽しみを見出すことが容易となる。しかし、自分の書いたコードに含まれる構文のみを学習することが多い。そのため、応用力の習得が困難となる。

上記した2通りの学習法から、プログラミング学習の流れは、解決型学習法の欠点である構文への理解を容易にし、早期にプログラミングへ移行可能なシステムが望まれる。

4. システム提案

本研究では、対象者を高校生、大学生のプログラミング初心者とし、対象言語をJavaに設定し、以下の特徴を有するシステムを提案する。

疑似体験によるプログラミング学習

物語を有した学習構成から目標を提供

動機付けの低下防止を狙い、視覚に訴える動作

を有するプログラムを作成

視覚を利用した構文理解の促進

構文理解にアニメーションを付加し、語彙の取

り違えによる理解の困難さを軽減

構文学習からプログラミングへ早期移行

指導者への学習過程の伝達

学生が構文理解のために行うクリック箇所及

び回数をデータベースに保存する。これを

者に閲覧させることで、学生が理解に苦し

んだ箇所を指導者が把握可能

5. 疑似体験

プログラミングは、目的に対し、様々な達成方法が存在する能動的活動である。ゆえに目標を提供し、達成感を得られる教材が必要である。

そこで、本研究ではプログラミング学習において、プログラムの役割を疑似体験させるような構成を付加する。構成要素を以下に示す。

プログラム

本学の研究室配属生と設定

疑似体験

配属生が研究室での活動において、プログラマ

として直面する状況を模倣することと定義

物語

配属生は研究に必要な基礎知識習得の一環として課題を提示される。課題を満たせない場合、研究室の先輩から指導を受ける。このような研究室での活動を物語として、プログラミング学習に付加

プログラミング題材

ツールとしての普及の高さから、学生が動作結果を想像し易く、処理方法の考案が容易なこと、また、視覚に訴える結果からの達成感を考慮し、ペイントツールを採用

6. システム構成

システム構成を図1に示す。学生はブラウザからサーバ上のHTMLを取得する。構文学習では、各单元毎にJavaScript制御されたFlashアニメーションが表示される。これにより、HTML内のクリックポイントに応じて、アニメーションの途中及び反復表示が実行される。プログラミング学習では、学生はブラウザから必要に応じて、JavaAppletを取得する。

学習過程で行ったクリックは、その箇所、回数がデータベースに自動保存される。この制御をCDMLが実行する。CDMLが新たに書き出したHTMLを閲覧で、学習者の学習過程を把握可能となる。

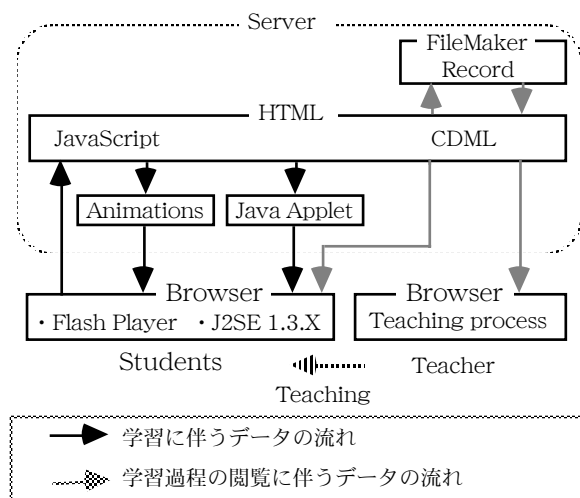


図1 システム構成図

7. 学習の流れ

学習の流れを図2に示す。

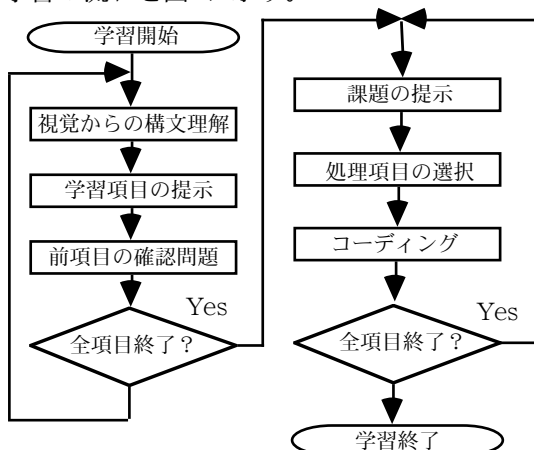


図2 学習の流れ

構文学習では、アニメーションとコードが同期したページを閲覧させ、基礎的な思考法を学習させる。その後、研究室の疑似体験が開始され、課題が提示される。学生は自らの処理法にて、コードを記述する。処理法の考案が困難な場合、物語上で、研究室の人間から処理例が提供される。更に考案が困難な

場合、学生は他の学生に意見を求める必要がある。その際、意見を求められた学生は、求めた学生の構文学習状況の閲覧を許可される。全処理項目を作成、結合し、終了となる。

8. 実行画面例

構文学習画面例を図3、ペイントツール例を図4に示す。構文学習画面では、学生が解説文の一文をクリックするとアニメーション、コードの注目位置及びその意味が逐次表示される。これにより、解説文を用いて、指導者が伝達したい心象、コードの注目箇所及び意味を学生にその都度確認させ、伝達する。

ペイントツールは、曲線の描画、フィールドの全消去及び部分的消去、太さ変更、色変更などの機能が描画スペースに反映される。また、現在使用中の機能名を表示する機能を持つ。

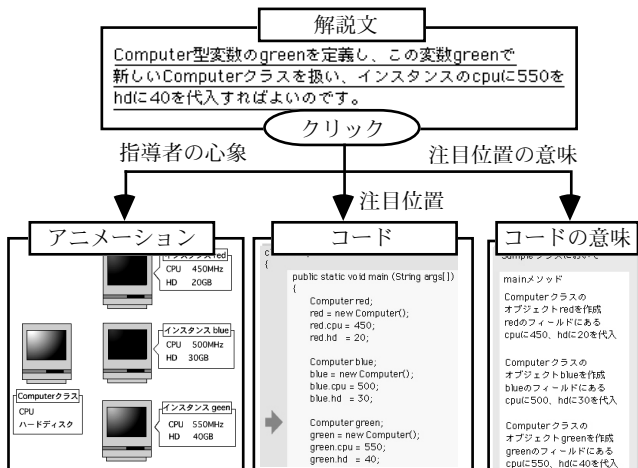


図3 構文学習画面例

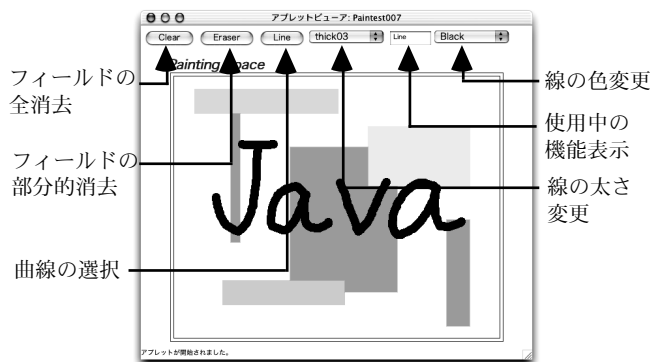


図4 ペイントツール例

9. おわりに

本研究で行った視覚からの構文理解は、解説文において、指導者が描いた心像、示すコードの位置、意味と、学習者が解釈したそれらを文章単位で確認可能となる。よって、構文理解の困難さを軽減すると思われる。また、疑似体験によるプログラミングは、学生に対して、研究室という特殊な環境下での物語を追わせる楽しさと、具体性を有する目標の提供をし、動機付けの低下を防止すると思われる。