

平成 30 年 6 月 12 日現在

機関番号：13801

研究種目：基盤研究(C) (一般)

研究期間：2015～2017

課題番号：15K04493

研究課題名(和文)プログラムのデバッグ相当作業を母語で事前トレーニングする方法の提案と教材開発

研究課題名(英文)A proposal of pre-training method for debugging-like work by mother language

研究代表者

太田 剛(Ohta, Tsuyoshi)

静岡大学・情報学部・教授

研究者番号：40213730

交付決定額(研究期間全体)：(直接経費) 500,000円

研究成果の概要(和文)：「手順的な自動処理」を学ぶためのトレーニング方法を開発した。このトレーニングでは、学習者が処理方法を母語で記述し、それを他者に実施してもらった結果のフィードバックを受けて修正することを繰り返す。初年度は、他者として同級生を割り当てる方法を開発した。次年度にこれを実践した結果、人が無意識に行ってしまう「善意の解釈による記述漏れの補填」「評価基準のゆらぎ」が、トレーニング上好ましくないことを発見した。これを解消するため、最終年度には他者としての簡易自然言語処理システムを開発した。

研究成果の概要(英文)：I developed a training method to learn how to describe "procedural automatic processing". In this training, a student repeats that describing the processing method in his/her mother language, and correcting it by the feedback of the result that carried out by a training-partner. In the first year, I developed a method assigning classmates as a training-partner. As a result of practicing this in the next year, I discovered that "too much supplemented interpretation for unsufficient description" and "fluctuation in evaluation criteria", that human unconsciously do, are undesirable for this kind of training. In order to solve this issue, I developed a very simple natural language processing system as a training-partner in the final year.

研究分野：情報科学

キーワード：教育学 情報科学 プログラミング教育

1. 研究開始当初の背景

コンピュータがどこにでもある世界を見据え、自分ができるタスクを他人あるいは機械にもできるように、手順を精確かつ詳細に記述する能力を、学校教育の中で身に付けさせることが求められるようになってきている。これを実現するべく、全世界的に、小中高校の授業の中でプログラミング的(以下、情報処理学会の提言に倣って「手順的な自動処理」と表記する。「プログラミング」は伝統的意味に限定して用いる。)教育活動を行った実践報告が数多くされており、個々の活動において成果も上がりつつある。

一方、大学における一般教養としての情報系科目においてプログラミング教育を行うことが、全世界的にあたりまえのこととなりつつある。しかしその現実、性別、年齢、学部学科に関わらず(つまり、情報系学部であろうと、いわゆる文系学部であろうと)、その約半数の学生は入門的な授業でさえも合格できない状況にあるとの報告がある。

研究代表者は、「手順的な自動処理」の考え方とスキルは、読み書き計算と同様に、人間誰もが身に付けられるものであると信じている。それでも約半数が脱落してしまうということは、明らかに現在の方法論に何等かの問題があることを示唆している。しかしながらその一方で、現在の方法論であっても、できる学生は「すんなり」と理解してどんどん伸びていくのもまた現実である。

とすれば、手順的な自動処理の考え方を学ぶ前提となるべきスキルが何か存在し、そのスキルを既に身に付けている学生は支障なく学ぶことができるものの、当該スキルを身に付けていない学生が途中脱落してしまう...と考えるのが、最も妥当な解釈であろう。そしてそのスキルは、学校教育の場では、その必要性がまだ認識されていないために意識してトレーニングすることがなく、学校以外のどこかで身に付けた者とそうでない者とが分かれるような「何か」であろう。

研究代表者は、大学におけるプログラミング教育の経験を通して、このスキルにかかわる仮説をいくつか持つに至った。そして、この仮説に基づいて挑戦的萌芽研究(H24~H26)「プログラミング教育を始める前の事前トレーニング法の提案と教材開発」(課題番号:24653271)を実施した。そこで得られた成果は、プログラミングを割とすんなり習得できる学生とそうでない学生との差は、日本語による「手順的な自動処理」記述にあたって、「相手に誤解されない表現」を意識して使えるかどうかにあるらしいことがわかってきた。つまり、解くべき課題を理解し解決することを自分の頭で行うことの他に、その解決方法を相手のレベルに合わせて伝える

表現や説明の詳細を相手に合わせて選ぶ能力がポイントであるらしいことがわかってきた。

2. 研究の目的

先行した挑戦的萌芽研究においては、伝えるべきことを「手順的な自動処理手順としてひとまず1度書きくだす」ところまでしか扱えていなかった。この後、書き下した処理手順は、相手によって解釈され実施される。ところが多くの場合、相手は自分が想定しなかった解釈をし、実施をしてしまうものである。その実施結果を観察し、記述のどこに問題があったかを推測して修正する作業が、現実にはこの後必要となる。プログラミング用語で言えば、コンパイル可能なソースコードがとりあえず書けたという状態には達しているが、デバッグ作業は未だこれからという状況にとどまっているのである。本研究では、この観察・修正の部分についてのスキルを、日本語による記述レベルでトレーニングする方法論と、そのための教材を提案することを目的とした。

3. 研究の方法

研究方法のポイントは、「プログラミングを学ぶ前提となるスキルは、学校教育の場でいまだ必要性が認識されておらず、そのためトレーニングもされていないものであり、学校以外のどこかで身に付けた者とそうでない者とがいる」という解釈にある。研究代表者は、このスキルに関わることとして、次の点を強調したい。

プログラミングは、仕事の進め方をコンピュータに教える活動である。

学習者から見たとき、学校は知識や課題の解き方を先生から教えてもらう場であって、自らが教える場ではない。友達に勉強を教えたり、リーダーとなってグループをまとめたりする機会をもつ児童・生徒は、一部にすぎない。このような機会は主に学校の外にある。具体的には、兄弟姉妹、地域やボランティアの活動の中で、自分と同年代あるいはより年下の子の宿題やめんどろを見る活動の中に存在しているのである。こういった機会を日常的に数多く持つ児童・生徒は、少子化の流れもあって、おそらく全体の半数以下であろう。これは、プログラミングをすんなり理解できる大学生が約半数であることと符合している。

一般に、誰かに何かを教える際、教授者の活動は次の通りである。まずは、学習者の質問等を参考にして、対話を通して初期学習者モデルを構築する。その学習者モデルを評価して欠如知識や誤解を見つけ出し、その原因を推測し、修正する方法を考えて実践する。その後、さらなる対話によって学習者がどう変化したかを評価してモデルを更新し、再び修正方法を考えて実践する。これを、学習者が理解したと判定するまで繰り返す。

この活動は、プログラミングと極めてよく似ている(表1)。教授活動では、意思を持った「人」を相手にし、曖昧さを持つ自然言語を用いているため、学習者モデル構築と評

表 1 教授活動とプログラミングの関係

教授活動	プログラミング
初期学習者モデル	初期プログラム
対話による評価	プログラム実行と結果観察
欠如知識や誤解を推測して修正	デバッグ(虫取り)

価が難しい。この点プログラミングは、モデルが完全に定義された「機械」を相手とするから、その点では楽と言える。その一方、初学者にとっては、自然言語を使えない難しさがある。

以上をふまえて、本研究においては次の考え方を基本とした。

- ・ 2人の学習者をペアにして、相互に教えあう体制のトレーニングとする。
- ・ 教えあう内容(教材)について難易度を調整して、学習者のレベルに合わせる。
- ・ 自然言語を用いる(コンピュータとプログラミング言語は用いない)。
- ・ 学習者は、お互い相手の意図を推測せず、伝えられた通りにする。上級レベルでは、推測したうえで、伝えられた内容を逸脱しない範囲で、敢えて相手の意図の逆をつくのも面白い。

これにより、自分の意図通りのことを相手にしてもらうために、用語を定義して相互理解することの必要性、文型を限定して誤解の恐れを極力抑える必要性、言葉を発する順番や係り受け関係、どの程度まで細かく指示しなければならぬか等、これまでは意識されていなかった、プログラミングを始める前に体得しておくべき様々なことについて実践的に学ぶことができることを狙った。また、小学生用の教材から高校生らしい教材まで、様々なレベルや難易度のものを揃えることで、徐々に高いスキルを身に付けていくことができる。なお、この方法論は、現在様々な学校で行われている「手順的な自動処理」教育との間で相互干渉して不具合を生じるものではない。

4. 研究成果

4.1 学習者ペアによる事例収集

手順的な自動処理をどのように記述すると、他人は解釈や実施を誤るのか、その事例を集めてパターンを分類することをまず行った。具体的には、プログラミングの授業において、先行した挑戦的萌芽研究において開発した教材を活用し、曖昧な記述や誤読・誤解事例を、2年間に亘って集めた。具体的には、漢字「本」を書く手順を、小学1年生を想定した相手に、ことばだけを用いて伝達・記述する課題である。

ここでは、(1)学生には何の示唆も与えない状況のもとで、ある形状のものを描くための日本語による処理記述文書を書き起こし

た第1版記述、(2)その記述を他人が読んで実施したときに描かれたもの、(3)描かれたものを見て、何を誤読されたか考えて修正した第2版記述、(4)第2版を先の同一人物が読んで実施したときに描かれたもの、(5)その結果をみて修正した第3版記述、を約80人分収集した。そしてこれらの事例から、多くの学生が陥った(1)共通の誤り易い箇所、(2)誤読され易い表現、を抜き出して分類した。

4.2 プログラミング行動の履歴収集

本学が有する既存システムを用いて、プログラミングの授業における学生の行動(プログラム編集、コンパイル等のイベント履歴)を、2年間に亘って収集した。

4.3 両収集データの分析

日本語による処理手順表記による事例とプログラミング行動履歴との間の関係について分析した。その結果は、2018年3月に情報処理学会全国大会において発表した。その特徴的な結果について以下に述べる。

日本語による処理記述文書に関して、第1版記述を、先行した挑戦的萌芽研究において開発した教材における基準に従って採点した。ここでは、一定基準以上の得点を得た学生達を上位グループ、そうでない学生達を下位グループと呼ぶことにする。

また、プログラミング行動に関して、「ユーザから受け取る入力値が1の場合は...、2の場合は...と動作するプログラムを作れ」という課題において、本来なら3つの場合(1,2,その他)に分けてif...if...elseと組み立てなければならないところを、1と1以外に分けて単純にif...elseで片付けてよしとしてしまった学生が相当数いた。if...if...else型の学生達を成功グループ、if...else型の学生を失敗グループとここでは呼ぶことにする。

分析の結果、上位グループと下位グループとは、成功グループと失敗グループへの分布の仕方が明らかに異なっており、5%水準で有意であった($\chi^2(1)=3.97, p<0.05$)。また、採点基準に含まれているある特徴量が、成功グループ・失敗グループにおいて異なることも判明し、これもまた5%水準で有意であった($t(35)=4.67, p<0.05$)。

これらのことから、このような課題(日本語による処理手順記述、プログラミング双方)においては、現在注目している対象が、他の対象との間でどのような関係にあるのかを認識して、それをきちんと記述できているかどうかという点が、成功・失敗、ひいては習熟度合いに影響を与えているのではないかとと思われる。

4.4 本方式の欠点と解決策

研究を進めていくうちに、本研究が当初想定していた「2人の学習者ペアによるトレーニング」の有効性に限界があることが判明し

た。

人は相手の意図を推測し、曖昧な記述や記述されていない部分について、無意識のうちに、善意に解釈して補ってしまう。これは、プログラミング作業における「書いてある通りそのまま実行する」コンピュータとは全く異なる人間の特性であり、日本語による処理記述文書の第1版から第2版、第3版と版数が上がるにつれ、トレーニング効果を維持することが難しくなるという大きな問題をはらんでいることがわかったのである。

さらに、人が行う評価にはゆらぎがあることも判明した。例えば、第1版を読んだときには記述誤りや漏れに気づかなくても、第2版のときに気づくことがある。あるいは、Aさんは気づいても、Bさんは最後まで気づかないという個人による違いもある。これらもトレーニング上好ましいことではない。

これらを解消するには、ペアの一方に、「いっさい推測をせずに、書いてある通りに実施する」ことを徹底することが考えられ、実際その方法で実施してみもした。しかしながら、その学生自身も学習者なのだから、日本語記述の不十分なポイントがどこにあるかを理解しているわけではない。したがって、どれだけ徹底してみても、「書いてある通り」にはできず、日頃の習慣を知らず知らずのうちに適用してしまい、善意の解釈および評価のゆらぎが生じてしまう。特に版数が上がるほど、前の版までの経過が頭に残っており、善意の解釈を撲滅することが難しくなる。

かといって、ペアの一方を経験のある教員に任せるのでは、教室におけるリアルタイム学習がほぼ不可能となる問題点がある。

そこで、日本語による処理記述文書を入力すると、簡単な日本語解析をして、文字通り、書かれている通りに線画を描画してくれるツールを開発することにした。文書例は豊富に収集してあったので、記述文例を分類し、本格的な自然言語処理を用いなくても、パターンマッチングによる方法で標準的な記述の相当部分をカバーできることを机上で確認した後、システム化した。

このシステムは、文書の編集作業をすべて履歴として管理できる。そのため、文書を記述し、システムに描画させ、その結果を目で確認し、どの指示に問題があったかを確認して、文書に修正を行う一連の作業をすべて記録することができる。処理の指示対象が線画に限られるという制限はあるが、後続の研究を進めていく上での、基盤となる有用なツールとなった。

5. 主な発表論文等

〔学会発表〕(計 1 件)

鵜飼凌、太田剛：“作業手順伝達文書の記述能力とプログラミング修得過程との関係調査に関する研究”，情報処理学会第80回全国大会 7J-03 (2018)。

〔その他〕

4.4節で説明したソフトウェア(名称未定)

6. 研究組織

(1)研究代表者

太田 剛(OHTA, Tsuyoshi)

静岡大学・情報学部・教授

研究者番号： 40213730