

## 学習項目ごとの Target Badge 認定による学習者の動機付けの向上

Improving Motivation through Badge Certificate for Each Learning Item

松本 峻<sup>†</sup> 板戸 陽子<sup>‡</sup> Dinh Thi Dong Phuong<sup>†</sup> 島川 博光<sup>†</sup>  
Ryo Matsumoto Yoko Itado Dinh Thi Dong Phuong Hiromitsu Shimakawa

### 1. はじめに

情報系の学部の多くでは、学習者は講義や演習などでプログラミングを課している。一般的なプログラミングの演習授業では、与えられた課題を解決するプログラムを書く演習が行われており、基礎的なプログラムの書き方を身につけさせている。しかし、学習者により学習への動機づけや既に獲得している知識の幅は異なるため、同じ課題を与えても簡単に突破する学習者もいれば、困難を感じ挫折してしまう学習者もいる。

本論文では、プログラミングに必要な学習項目ごとに学習者の学習結果を評価する手法を提案する。課題を解くことによって学習者が十分に理解している学習項目を判断でき、学習者が自己の学習を内省するのに役立つ。

### 2. 自己の学習を評価する仕組みの必要性

#### 2.1 大学におけるプログラミング教育の課題

大学入学時には、学習者ごとに学習意欲のばらつきがあり、知識量も個々人で大きな差がある。さまざまな背景をもった学習者が存在するにもかかわらず、一斉授業で講義が進むことが多い。そのため、学習者一人一人にあった最適なカリキュラムになっておらず、毎年、プログラミングに挫折する学習者は後を絶たない。

挫折する要因のひとつとして、実力に見合わない困難な課題を与えられていることがあげられる。学習者は授業で知識を得たからといって、プログラミングの技能が身につけているわけではない。身につけた技能を遥かに超えた課題が学習者に与えられており、自分には向いていないのではないかと考え、挫折してしまう学習者が少なくない。これにより、さらなる学習への学習意欲やプログラミングに対する有能感が失われている。

#### 2.2 自己効力感と達成動機の関係

自己効力感とは、外界の事柄に対し、自分が何らかの働きかけをすることが可能であるという感覚である [1]。

特定の分野に対する自己効力感を specific self-efficacy と言い、プログラミングという特定の分野に対する自己効力感が必ず存在する。自己効力感の源泉となるものはさまざまな要因があげられるが、プログラミング分野においては達成体験がもっとも重要であると考えられる。なぜなら実際にコードを書いて動かすという機会が、プログラミングの授業を受けた後も多く登場するためである。プログラミング教育においては、適切な課題を設定すれば、その達成により達成体験が得られると考えられる。

自己効力感とは達成動機に帰属する。自分には才能があるからプログラミングができる、もしくは自分に才能がないからプログラミングができないのだと思う学習者と比べ、自己の学習への努力が反映されていると考える学

習者のほうが達成動機が高い。このような状況の学習者は、さらなる学習へ強い動機付けを持つようになる。

#### 2.3 自己調整学習と自己効力感の関係

自己調整学習とは、学習者が動機づけ 学習方略 メタ認知 の3要素において自身の学習過程に能動的に関与していることと定義される。[2] 具体的には以下の流れに沿って学習者が主体的に学習することである。

1. 学習者が動機付けにより学習を計画する。
2. 最適な学習方略を選択し目標を遂行する。
3. 学習結果に対して学習者自身が評価を行う。
4. 評価を元に自己の学習状態を認知する。
5. 更なる学習への動機づけにつなげていく。

指導者ではなく学習者が自らの学習結果を評価する過程が重視される。数学における先行研究において、自己調整学習を行えている生徒は自己効力感が高く、テストの成績も良いことが発見された [3]。これは数学に限らずプログラミング教育においてもあてはまるであろう。

現在の大学のプログラミング演習においては、ひとつの課題に対して複数のチェック項目が存在し、教員や補助員などとの問答によって、学習者の理解度を測っている。教員が学生を評価するための支援機構として文献 [4] が挙げられる。しかし、これらチェック項目は教員が学習者の理解度を判断するための指標であるため、プログラミングを初めて習う学習者にとって、自己の学習を内省するのに役立っていないと考えられる。よって、自己調整学習におけるメタ認知の部分において、学習過程に能動的に関与できず、次なる学習への動機付けや、その計画と遂行に大きな支障をもたらしている。

### 3. TargetBadge での項目ごとの到達認定

#### 3.1 学習項目ごとの評価

本論文では課題ごとの評価ではなく、プログラミングに必要な学習項目ごとの評価方法を提案する。本手法では、学習者が課題を回答し、その結果を基に学習者の身につけているプログラミング技術を判断する。ある学習項目に対して、一定の到達目標に達すると Target Badge が認定賞として付与される本手法の全体図を図1に示す。

学習者の項目達成状況はスキルプログレスで表示される。スキルプログレスは、表頭に学習項目、表側に達成度を持つ表である。これによりどの項目まで習得したかが一目でわかるようにする。学習者が自己の学習を評価するのを容易にし、次の学習に活かすことを促し、学習者の自己効力感やそれに帰属する達成動機を増大させる。

ひとつの課題を解くのに必要な学習項目はひとつとは限らず、広域に及ぶ。例えば、フィボナッチ数列の  $n$  番目を求めるという課題には、「関数の作り方」「再起」という項目が大きく関わり、「戻り値」「引数」「関数の呼び出し」といった項目も関わってくる。また、「if文」や

<sup>†</sup>立命館大学情報理工学部

<sup>‡</sup>立命館大学大学院情報理工学研究科

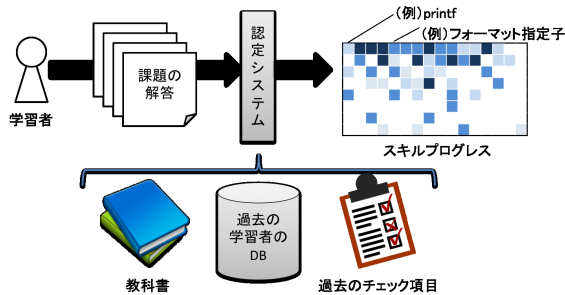


図 1: 手法の全体像

「printf による入出力」という項目も小さいながら関わってくる。スキルプログレスはこれらの関連を一覧で示るので、学習者はひとつの課題を解くのに必要な学習項目を事前に知ることができる。

さらにスキルプログレスは、達成されている学習項目と達成されていない学習項目を、各学習者に向けて表示するので学習者が複数の課題を解くことによって、どの学習項目に対してどの程度到達できているかを判断することができる。

### 3.2 スキルプログレスの作成

学習者に与えられる課題すべてに、その課題を解くのに必要な学習項目をあらかじめ分析しておく必要がある。Target Badge のスキルプログレスは授業に用いられた教科書、過去の学習者の学習経過を保存したデータベース、課題ごとに設けられたチェック項目の 3 つを基に作成する。作成の手順は以下のとおりである。

#### 1. 学習項目のリストアップ

大学で用いられた C 言語の教科書 3 冊を元に学習項目をリストアップする。学習項目は、細かい粒度にまで分解される。

#### 2. 課題を解く経路を求める

課題間の相関を有功グラフにした例を図 2 に示す。課題 B が課題 A を解くのに必要な前提知識であるとき、課題 B から課題 A への有向経路が持つコストを  $cost(B \Rightarrow A)$  とする。過去に出題された演習課題にアソシエーション分析をかけ、課題 A を解いている人が課題 B を解いているときの支持度を  $supp(A \Rightarrow B)$ 、確信度を  $conf(A \Rightarrow B)$  とする。出題された課題の総数を  $M$  とすると、 $cost(B \Rightarrow A)$  は次の (1) の式で表すことができる。(1) により現れる完全グラフに対し、もっとも支持度の高い結論を始点、もっとも支持度の低い結論を終点とし、ダイクストラ法で経路を探索する。このとき、一度も探索されずに残った線を削除することで、課題間の関連度が予測できる。

$$cost(B \Rightarrow A) = \left( \frac{1}{supp(A \Rightarrow B) conf(A \Rightarrow B)} \right)^M \quad (1)$$

#### 3. チェック項目と学習項目の関連付け

課題が持つチェック項目と、学習項目を関連付ける。これにより、課題を解くのに必要な学習項目が判明し、学習者の課題到達状況が判断できるようになる。

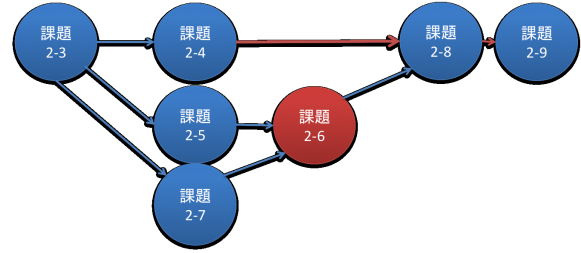


図 2: 課題間の相関グラフ

### 4 達成度の計算

上記 1 で得られた学習項目を表頭におき、上記 3 で得られた課題と学習項目の関連に基づき、ある課題が解けた場合の各学習項目に対する達成度を計算する。達成度の表現においては、2 で示したコストを用いる。

### 3.3 Target Badge を導入することによる効果

自分の項目達成状況と課題が問うている学習項目を比較し、自分にとってどの程度の難易度を持つかを知ることができるため、自分の技能の過不足が一目でわかるようになり、自己で学習の評価ができるようになる。

学習者ごとに、得意とする分野、苦手とする分野は異なり、学習者自信がどの分野で躓いているかを判断することができる。例えば、関数に対しての理解が弱い学習者と、配列に対して理解が弱い学習者とで、復習すべき分野や学習方法が異なる。

この手法により、漠然とした「プログラミングが苦手である」という認識から、自分は特定の分野の理解が進んでいないから課題を解くのが困難なのだ、学習者自身が気づくことができる。自ら学習の評価ができるようになると、次なる学習への学習方略が立てられる。自らの学習計画を遂行することにより学習者の自己効力感が向上し、達成動機に帰属する。

### 4. おわりに

本論文では、学習項目ごとの評価による学習者自身の学習の評価を促す手法を提案した。今後は当手法の検証のための実験を行う予定である。

### 参考文献

- [1] 無藤 隆, 森 敏昭, 遠藤 由美, 玉瀬 耕治, "心理学" 有斐閣, pp191-pp211, 2001 年
- [2] 伊藤 崇達, 神藤 貴昭, "自己効力感, 不安, 自己調整学習方略, 学習の持続性に関する因果モデルの検証" 日本教育工学会論文誌 27(4) pp377-385, 2003 年
- [3] 松沼 光泰, "テスト不安, 自己効力感, 自己調整学習及びテストパフォーマンスの関連性" 教育心理学研究, 2004, 52, pp426-436
- [4] 小西 達裕, 鈴木 浩之, 伊藤 幸宏, "プログラミング教育における教師支援のためのプログラム評価機構" 電子情報通信学会論文誌, 2000, 83, pp682-pp691