N-010

プログラミング演習問題の技術要素に基づく理解度の把握

Recognition of Individual Understanding Using Technical Elements of Programming Exercises

小柳 順一† **島川 博光**†
Junichi Koyanagi Hiromitsu Shimakawa

1. はじめに

現在のプログラミング教育では，プログラムの実行結果とソースのみが評価される場合がほとんどであり，学習内容ごとに細かく評価されることはない．そのため学習者は自身の理解度を把握することができず，学習内容の適切な復習も行うことができない．最悪の場合，落ちこぼれる学習者が発生してしまうこともある．

本論文では，これらの問題点を解決するために，C プログラミングを例にして，学習内容を細かく分割し，そのひとつひとつの理解度を評価するための手法を提案する．提案手法では学習者は細かな評価よりも課題ごとの総合評価を気にする傾向があるので，内容ごとの細かい評価から課題の総合評価を算出できるようにしている．また，複数の教員が分担して指導を行う場合でも，同じ基準で評価するので，教員により評価に差が生じることはない．

学習者が自らの理解度を認識しやすくするため，理解度をグラフィカルに表示するツールを実装する．このツールでは，学習内容ごとの理解度の履歴と課題ごとの得点履歴がグラフで表示され，評価理由についても表示される．また，教員の評価にかかる負担を軽減させるために，教員用の評価入力画面も実装している．

2. プログラミング教育の現状

現在，大学などで行われているプログラミング教育では教員が課題を出題し，学習者はそれに対するプログラムを作成して提出する．しかし、ほとんどの教員は，実行結果とソースコードを評価するだけであり，それだけでは各学習者の詳細な理解度を評価することができない．習得すべき技術要素ごとに教員が細かく評価を行い，学習者に対して，その評価結果の詳細な説明が提示されれば，学習者も自らの理解度を把握できる．しかし，多くの学習者を受け持つ教員が，各学習者に対してこの作業を行うには，多大な労力と時間が必要であり，事実上不可能である．実際，学習した内容の復習は学習者自身に委ねられているので，学習者は自分の弱点をつかむことができないと適切な復習が行えず，学習内容を理解し切れていないまま取り残される．その結果，一部の学習者は落ちこぼれてしまう傾向にある．

プログラミング教育を支援するための研究は活発に行われているが，この問題については未解決のままである。たとえば，文献[1] では，教員が各学習の学習状況を把握するための手法を提案しているが，各学習者の弱点を見つけ出したり，学習者の詳細な理解度を把握するための研究はなされていない．

† 立命館大学　理工学部

3. 単元ごとの理解度の把握

3.1 技術要素

文献[2]では C 言語の学習内容を基礎的な 8 つの単元に分類して扱うことを提案している．各々の単元の学習内容は，さらに複数の学習項目に分割される．我々は図 1 に示すように，単元ごとの理解度を学習者に示すことを提案する．本研究では，この学習項目をプログラミングに関する技術的理解度を評価するための技術要素と捉える．

また，技術要素の理解度を評価するための観点を評価観点とする．たとえば，入力された 10 個の整数をリストを使ってソートする課題において「標準入力より入力された 10 個の整数を読み込む」といった事項が評価観点であり，これが実現できているかが採点者により調べられる．この調査により「書式付き入力」の技術要素が理解できているかどうかが評価できる．

3.2 技術要素の点数付け

技術要素の点数付けを行うためには，課題に含まれる技術要素を見つけ出す必要がある．しかし，この判断を採点者に任せると採点者の主観に左右されてしまい，複数の採点者が存在する場合には，採点者によって課題に含まれる技術要素の認識にズレが生じる可能性がある．

そのため，課題に含まれる技術要素は課題作成者が課題を作成するさいにあらかじめ提示しておくことにし，同時に評価観点も設定してもらうことにする．評価観点の点数の幅は評価観点ごとに異なる幅を持たせる．この点数の幅と採点基準についても課題作成者があらかじめ決定しておくものとする．採点者は課題に設定された評価観点ごとに，採点基準に基づいて点数を付ける．プログラミングの出来具合によっては，マイナス点になることもある．

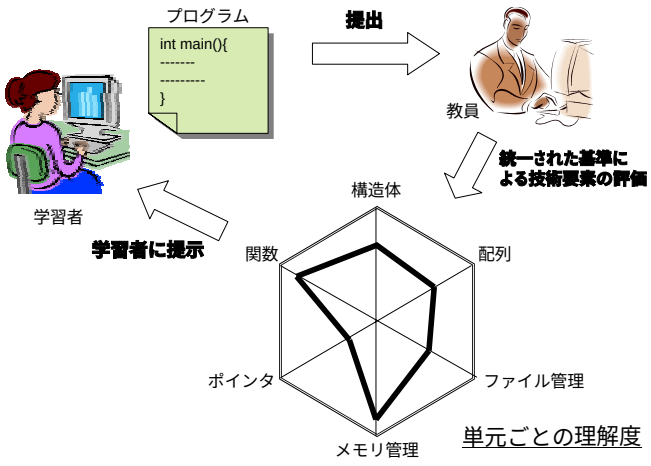

図 1：単元ごとの理解度の提示

3.3 理解度の把握

評価観点ごとに付けられた点数はそれに対応する技術要素が含まれる単元の得点に反映される．したがって，各単元の得点は課題が完成するたびに変化する．この単元ごとの得点履歴を学習者に提示することで自らの細かな理解度が把握できる．

しかし，学習者は単元ごとの細かな評価より課題ごとに技術要素を総合的に評価した結果を気にする傾向があり，単元ごとの得点履歴だけでは学習意欲は向上するとは考えにくい．そのため，課題ごとの得点を学習者に提示することも必要不可欠である．

そこで，課題ごとの総合評価を以下の式で求め，その履歴も学習者に提示する．

$$\text{総合評価}(\%) = \frac{\text{各評価観点に対して獲得した点数の総和}}{\text{各評価観点の配点の総和}} \times 100$$

課題ごとの得点だけでは，その学習者の特徴は特定できない．本研究では単元ごとの得点履歴と課題ごとの総合評価履歴を用いて，2 つの観点から学習者の理解度を考察する．課題ごとの総合評価が著しく下降している箇所について，単元ごとの得点履歴から下降している原因を特定する．結果として得られた単元が学習者の弱点だと判断する．

さらに，その単元の得点が下降している詳細な原因はソースコードを評価するさいの評価観点から推測することができる．それら 2 種類の得点履歴と点数が下降している理由を表示することで，学習者に自らの理解度を明確に把握させる支援が可能になる．

4. グラフィカルな理解度の表示

学習者の理解度をグラフィカルに表示することで，視覚的に得点の履歴を認識しやすく，理解度を把握できる．そこで，採点者用の入力 Web ページと学習者の課題ごとの総合評価の履歴と単元別得点履歴をグラフィカルに表示する．これらは Servlet と Applet を用いて実装する．

4.1 演習課題評価シート

採点者は Web ブラウザから課題作成者があらかじめ課題ごとに提示した評価観点に対して，採点基準に基づいて点数を入力する．評価観点ごとに点数の幅があるので，採点者はその幅の間で点数を付ける．このとき，点数の入力はスクロールで行い，採点者の入力の手間の負担を軽減している．採点時に評価観点に対するコメントも入力ができ，このコメントは学習者に理解度を表示するさいの評価理由の一部になる．

4.2 単元別得点履歴グラフ

単元ごとの得点については図 2 に示すように，学習内容を分類した 8 つの単元のうち 6 つの単元の得点グラフが一覧表示される．標準入出力と制御文の単元は，学習が進むにつれて得点が上がり続けることが予想できるので，得点の変化が大きいと考えられる 6 つの単元に焦点を絞っている．グラフ内のプロットをマウスでクリックするとそのときの点数に対する評価理由が表示される．

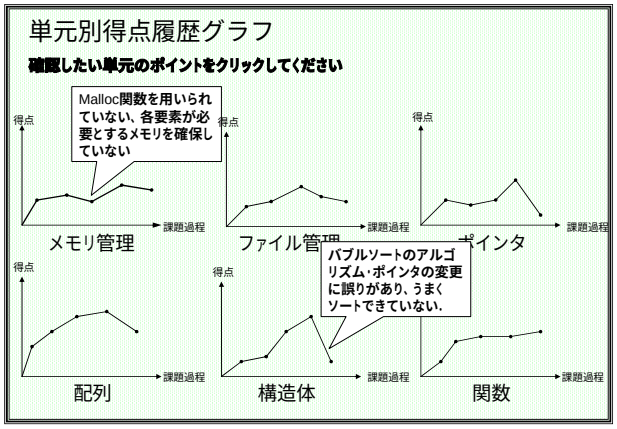

図 2：単元別得点履歴グラフ

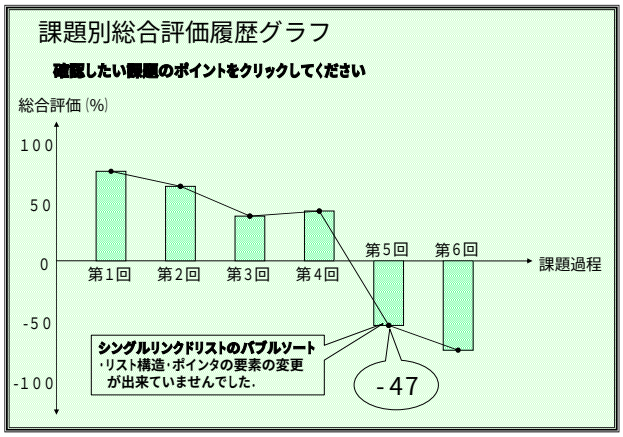

図 3：課題別総合評価履歴グラフ

4.3 課題別総合評価履歴グラフ

課題ごとの総合評価の履歴は，図 3 に示すように棒グラフで表示する．さらに，課題の総合評価コメントを表示する．グラフ内のプロットをマウスでクリックすると課題の総合評価に対する評価理由が表示される．

5. おわりに

本論文では，C 言語を例にして，学習内容ごとの詳細な理解度を評価するための手法を提案した．さらに，学習者の理解度の履歴を視覚的に表示するツールを実装した．

今後は本手法の有効性を実験により確かめる予定である．

参考文献

[1] 諏訪 正則，倉澤邦美，鈴木惠介，森本康彦，横山節雄，佐々木整，宮寺庸造，“プログラミング教育における学習履歴取得システムの開発”，第 2 回情報科学フォーラム（FIT2003）講演論文集，pp.583-584(2003).

[2] 田口浩，島川博光，石井篤，“個人の理解度に合わせた C 言語プログラミング教育支援環境”，*proc. of the 2nd International Conference on Creating, Connecting and Collaborating through Computing*, 日本語セッション，pp.60-63(2004)