Vol.2013-CE-120 No.1
2013/7/6

プログラミング演習の自動採点システムの評価法と進捗状況

内藤 広志[1,a)]　齊藤 隆[2]

概要：コンピュータ支援採点システムが学生のプログラムを評価する方法として，プログラミング課題の仕様を 4 つの種類に分類し，それに基づいて重み付けする評定法 WC を提案した．本方法による評定値と教員の評定値を比べると，教員の評定法はプログラム全体を見ている，重み付けを柔軟に変更していることに特徴があることがわかった．また，学習履歴データを解析することで，評定値と進捗の関連を調べ重み付けなしの評定法 C と比較をしたが，両者で大きな違いはなかった．教員アンケートでは，評定法 WC と評定法 C のどちらが妥当かは意見が分かれたが，本評定法を用いて進捗を表示するのは有用性が高いと評価された．

Grading Student programs for visualizing progress in classroom

Naito Hiroshi[1,a)]　Saito Takashi[2]

Abstract: To grade student programs in Computer-Aided Assessment system, we propose the method WC which is passed testing counts weighted by types of specifications for programming assignment. Comparing grade value between CAA system and teachers, teachers grade programs holisticly and change weights flexibility. By analyzing the association between grade and learning record data, method WC is a little better than not weighted method C. By questionnaire, teachers can not be concluded that either WC and C method are better but they think it's highly useful to display the progress using these methods.

1. はじめに

プログラミング演習では対象となる学生が多いため，大人数の演習教室を使って実施することが多く，学生アシスタント（TA）を含めた多くの人的リソースを必要とする．しかし，教員や TA が教室を巡回して学生の質問に答えるだけでは効果的な演習を実施するのは難しい．学生の進捗状況を把握し，多くの学生が犯している誤りを発見し，その原因を理解した上で，学生に注意をしたりヒントを与える必要がある．そのために教員はコンピュータの支援を必要としている．

プログラミング課題の評価する教員の負荷を軽減することを目的に，欧米や日本の大学で多くのコンピュータ支援評価（computer-aided assessment, CAA）システムが開発されて実際の演習で利用されている [1], [2]．CCA システムは，学生のプログラムが課題の仕様に従っているかを検査するために，ソフトウェア工学の様々なテスト技法を用いて学生のプログラムを自動または半自動で評価する．テスト技法には，プログラムを実行してプログラムの機能などを検査する動的な解析やプログラムを実行せずにソースコードを解析する静的な解析がある．

本学部では，C 言語のプログラミング課題を評価するための CAA システム Hercules を 2003 年度より開発し [3]，その後，C プログラムを検査機能を充実させるとともに，Java 言語で書かれた GUI プログラムの検査モジュール [4] や，XML 言語を用いるプログラムの検査モジュール [5] を開発してきた．Hercules は，100 人規模の演習教室が 6 部屋の演習科目で利用している．その機能として，1) 演習時間中に学生が作成中のプログラムを評価して進捗情報を生

[1] 大阪工業大学 情報科学部 情報メディア学科
Department of Media Science,
Faculty of Information Science and Technology,
Osaka Institute of Technology
[2] 大阪工業大学 情報科学部 情報システム学科
Department of Information Systems,
Faculty of Information Science and Technology,
Osaka Institute of Technology
[a)] naitoh@is.oit.ac.jp

成し，その情報を教員に提示するモニタリング機能と，2) 課題提出の締切後に学生が完成したプログラムを評価し，その結果を学生に提示する採点機能をもつ．各演習教室で指導する教員は，この進捗情報を見て誤りの多い課題に対しては補足説明やヒントを出したり，ある程度の正解率になった時点で解答例のプログラムを学生に示す．また，TA が誤りのある学生に直接指導することもある．このように，Hercules を利用することで進捗状況を把握するができ，大人数の学生に対して状況に応じた指導が可能となっている．

CAA システムは，テスト技法を用いて検査した結果を使って学生のプログラムの点数を付けたり，5 段階などの等級を評定する．プログラミング課題の評価法には全体的評価，分析的評価，特定要因評価があるが [6], [7]．全体的評価は，評価者の印象に基づく評価で，各等級の定義や解答例のプログラムを基準として評価する．分析的評価は，プログラミング能力を構成している下位項目を設定し，それらを個別に評価する．特定要因評価は，特定の評価項目を元に評価をおこなう．プログラムの評価の場合はソフトウェアの外部品質特性を評価項目とすることができる．点数や評定値は，プログラミング課題の到達度を学生が理解するためだけでなく，課題の難しさを評価したり，課題の進捗状態を視覚化するためにも重要である．多くの CAA システムは検査の結果を評定項目毎に重み付けして点数を計算する．BOSS システムのように教員が評価法をカスタマイズできるものもある [8]．しかし，学生は課題の意図の誤理解や誤字脱字も多く，関数名や出力メッセージを課題の指定どおりにならないことが多い．厳しい評価法にすると評定値が低くなり，学生のモチベーションが低下する．したがって，教員にとっても学生にとっても妥当な評定法が求められる．

本稿では，プログラミング課題の仕様の分類に基づいた評定法について提案し，その妥当性を様々な角度から分析する．

2. プログラミング演習の特徴

科目「C 演習 II」では，繰り返しや配列などの C プログラミングの基本を履修した学生に対して，次を学習目標としてプログラミング演習を実施している．

(1) 構造体，ポインタなどの高度な C 言語の文法と使用法を理解する．
(2) リスト，スタック，連結リストなどのデータ構造の実現法と利用法を理解する．
(3) 上記の学習内容を応用した 100 行程度のプログラムを作成できる．
(4) データ構造の抽象化を考慮してプログラムを作成できる．

これらの学習目標を達成するために，学習内容を基本的な項目に絞り，各回の学習テーマに対して学習目標（単元目標）を明確にし，難易度がスモールステップとなるような下位目標に分解し，下位目標ごとにプログラミング課題を設けている．そのため，少数の大きなプログラムを作成する代わりに多数の小さいプログラムを作成する課題を学生に課し，各単元で 20 行から 100 行ぐらいのプログラミングを 9 個ほど作成する．また，プログラミング課題の特徴として，限られた時間の中で効率的に学習目標を達成するため，ゼロからプログラムを作成するのではなく，説明用の例題プログラム (以降，ベースファイルと呼ぶ) を修正してプログラムを作成する問題が多いことである．

学生は完成したプログラムを電子メールや HTML フォームを使って提出するのではなく，学生のホームディレクトリ下の課題の説明文で指定されたファイルに保存する．学生のホームディレクトリは NFS サーバーに作成されているため，Hercules は学生が作成中のプログラムを参照し，評価できる．

3. プログラミング課題の仕様と評定法

3.1 課題の仕様と評定基準の定義

プログラミング演習で学生が作成するプログラム (以降，作成プログラムと省略) は，ある入力に対して指定された出力を生成するだけでなく，その課題で新たに学ぶ学習項目を満たしている必要がある．例えば，再帰的プログラミングを学ぶ単元では，指定の入出力機能をもつ関数を定義するだけでなく，それが再帰的呼び出しをおこなう関数として定義されている必要がある．

そのため，作成プログラムが満たすべき仕様を次の 4 つに区別する．なお，誤入力処理を入出力仕様と制約仕様から分離したのは，プログラミング経験の少ない学生には誤入力データの処理まで完璧に実現することを望むのは厳格すぎるからである．

表 1 課題の仕様と検査項目の種別

種類	説明	種別
学習項目	本課題で新たに学ぶプログラムの構成要素を表す．	■
入出力仕様	本課題の入出力データの対応を表す．	▲△
制約仕様	学習項目と入出力仕様を満たすために必要なプログラムの構成要素を表す．コンパイルも含む．	×△
誤入力処理	誤った入力データに対する処理を表す．	□

仕様の種類に対応して，表 1 に示す記号を使って検査項目の種別を指定する．作成プログラムは，関数名，引数や返却値の型が仕様とおりでなかったり，誤字や揺れのあるメッセージを出力する．正確さを求められる業務プログラムと違い，これらを一概に誤りとするのは学生のモチベーションを損ねるし，逆に，これらの誤りを学生に通知しな

課題6

work16base.cは、配列を使用して、入力データの平均値を計算するプログラムです。work16base.cを参考にして、次のプログラム work16.c を作ります。

1. 学習項目
   1. 配列に値を代入し、その値を参照する。
   2. for文またはwhile文を使って、配列のすべての値を参照する。
2. 課題の仕様
   1. scanf関数とwhile文を使って、EOFまで整数を配列 data に読み込みます。
   2. 配列 data に格納されたデータの最大値を求め、実行例のように出力します。
   3. 配列dataの定義は次のとおりです。main関数に定義します。

```
int data[100];
```

3. 注意事項
   A. 配列、for文、while文を使用します。
   B. main関数以外の関数を定義してはいけません。
4. 実行例は次のとおりです。コマンド行で**記号 <**（入力リダイレクション）を使うと、ファイルよりデータを読むことができます。ファイルdata5.datに対しては何も出力しません。

```
% ./work16.exe < data1.dat
11個のデータの最大値は99です
% ./work16.exe < data2.dat
6個のデータの最大値は74です
% ./work16.exe < data3.dat
4個のデータの最大値は3です
% ./work16.exe < data4.dat
1個のデータの最大値は43です
% ./work16.exe < data5.dat
```

5. work16.c ファイルは ~/kadai/padv13/1ec01 ディレクトリに保存しなさい。

図 1 プログラミング課題の例

いのも正しいプログラムを作成しようというモチベーションを高めない．そのため，このような誤りに対する検査項目には種別「△」を指定する．これら以外に，ベースファイルから変更のない作成プログラムは種別「写」を指定する．

図 1 は，配列の参照・代入を学習項目とする課題 6 の例である．制約仕様は注意事項として記述している．

図 3 に課題 6 の解答例のプログラムを示す．

3.2 評定の等級の定義

学習項目などの仕様を満たしているかを元に，作成プログラムの評価を次の 5 段階の等級にで評定する．ただし，5 と 4 が合格で，その他は不合格とする．「写」は 1 とし，ファイルがない場合は 0 とする．

表 2 5 段階の等級の定義

等級	説明
5	学習項目，入出力仕様，制約仕様を満たす．ただし，幾つかの誤入力処理を満たさない場合もある．
4	学習項目と入出力仕様を満たすが，幾つかの制約仕様を満たさない．
3	学習項目とその他の仕様を満たす度合いが十分に高くない．
2	学習項目とその他の仕様を満たす度合いが低い．
1	学習項目とその他の仕様を満たす度合いが非常に低い．

この 5 段階の等級と仕様との関係をまとめると表 2 になる．各等級で許す誤りを種別の記号で示す．

表 3 に示すように等級 1 から 3 は仕様を満たす度合い（点数）によって評定する．等級 5 と 4 は点数の他に表 3 に示す仕様に関する誤りがないという条件を追加している。本方法 (評定法 WC と呼ぶ) では，作成プログラムが誤りを検出しなかった検査項目 (パスした検査項目と呼ぶ)

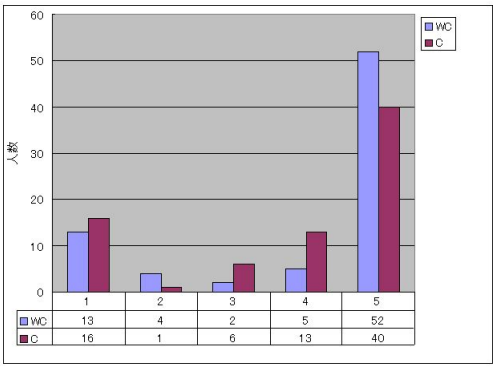


図 2 評定法 WC と C の評定結果の比較

の個数に種別ごとの重み付みをかけた値を使用する．重み付けの値は，■，×，▲，□，△が 4，2，2，1，1 である．例えば，パスした検査項目が■，×，▲，□，△をそれぞれ 5，4，3，2，1 個あった場合は 37 となる．ただし，課題によって検査項目の個数が異なるため，課題の解答例プログラムを 100 点とし，ベースファイルを 0 点として正規化して点数を計算する．なお，誤りの個数を使用しないのは，コンパイルエラーの場合に実行出力の検査ができないためである．

評定結果の比較のために，パスした検査項目の個数を点数化し，表 3 の点数の条件だけで評定する方法 (評定法 C と呼ぶ) も実施した．評定法 WC と C の評定結果を図 2 に示す．評価 5 の学生数が評定法 WC の方が多く，検査の種別によって重み付けすることで，誤りが軽症の場合は評価が高くなっている．また評価 5 と 4 の合計を比べると，評定法 WC では 57 名で評定法 C では 53 名となり，評定法 WC の方が誤りの程度によって評価を甘くなっていることがわかる．つまり，この課題の検査項目ではパスした検査の個数だけで評価すると，評価が厳しくなっている．

表 3 5 段階の等級と仕様との関係

等級	学習項目	入出力仕様	制約仕様	誤入力処理	点数
5	誤りなし	誤りなし	誤りなし	□	90 点以上
4	誤りなし	誤りなし	×	□	80 点以上
3	■	×	▲	□	70 点以上
2	■	×	▲	□	60 点以上
1	■	×	▲	□	60 点未満

4. 教員による評定法の検証

3.2 章で定義した等級の基準に基づいて，図 1 の課題 work16 の検査項目を作成し，316 名の学生の演習終了時点における作成プログラムを採点したところ，164 名の作成プログラムに誤りを検出した．この課題は，「繰り返しと配列」を学習テーマとする第 1 回の単元で特に誤りが多かった課題である．テストケースは，中間に最大値がある，先頭に最大値がある，末尾に最大値がある，入力データが 1

```
#include <stdio.h>
int main(void) {
  int data[100];
  int N, x, i, max;
  i = 0;
  while (scanf("%d", &x) != EOF) {
    data[i] = x;
    i++;
  }
  N = i;
  if (N > 0) {
    max = data[0];
    for (i = 1; i < N; i++) {
      if (data[i] > max)
        max = data[i];
}
    printf("%d 個のデータの最大値は%d です\n", N, max);
  }
  return 0;
}
```

図 3 課題 6 の解答例のプログラム

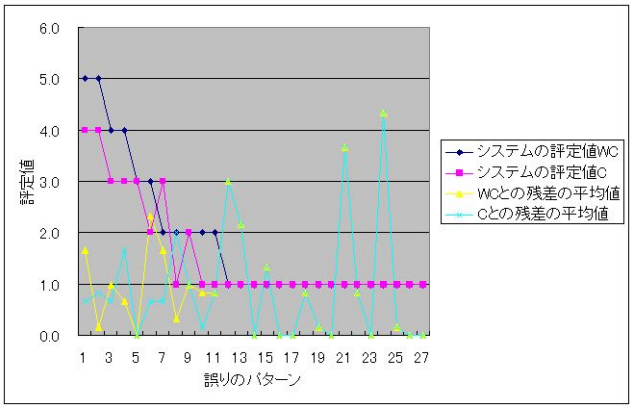


図 4 システムの評定と教員の評定との残差

個だけ，入力データが 0 個の 5 個である．検査項目の個数は，■が 6 個，×が 6 個，▲が 16 個，□が 1 個，△が 9 個，写が 2 個である．誤りの組み合せが異なるものを選ぶと 27 種類のパターンがあった．特に多いエラーは for 文の初期化処理と継続条件の誤りパターンである．

本評定法が妥当であるかを評価するため，本演習を担当する 7 名の教員に 27 個の作成プログラムのソースプログラム，採点結果 (誤りの検査項目のリストと実行出力)，システムの評定値を見てもらい，特定の評価法を教員に強制せず，各教員に自身の評定値を自由に点けてもらった．また，システムの評定値と自分の評定値が異なる場合は理由も明記してもらった．7 名の教員が採点上で重視する項目は異なり，評定値は異なる場合が多い．全体的に甘く評定する教員と辛く評定する教員が存在する．

システムの評定値と教員の評定値と差を概観するために，図 4 に評定法 WC と評定法 C による課題の評定値と各教員の評定値との残差の平均値を示す．パターン 12 以降は評定値 WC と評定値 C が同じ 1 のため，両者の残差は同じとなっている．

本課題では，while 文を用いて，標準入力から読み込んだデータを配列に格納し，次に for 文を用いて配列の最大値を計算するプログラムを作成することを意図している．入力データの最大値を計算するだけでなく，for 文を使ってすべての配列要素を参照できることを学習項目としている．

残差が大きく異なったのは，パターン 8, 12，13，21，24 である．それぞれの原因を調べると次のようになった．

(1) パターン 8 は，最大値が先頭にあるテストケースと入力データが 1 個だけのテストケースが誤りとなっている．

(2) パターン 12 は，1 つの for 文で最大値を計算している．配列の全要素の代入を while 文で，参照を for 文を用いておこなうことを学習項目とする課題であるが，教員は課題の説明文がその点を厳密に書いていないと考え，最大値を求めるプログラムとしては正しいと考える教員が多く，システムの評定値が正しくないと考えたようだ．

(3) パターン 13 は，for 文の代わりに 1 つの while 文を用いて最大値を計算しているプログラムである．パターン 12 と同様に課題の説明文が厳密でなかったため，教員の評定値がシステムの評定値と異なった．

(4) パターン 21 は，最大値は正しいが入力データ数の出力が正しくない作成プログラムである．出力データの検査では，出力される数値データの検査をおこない，それらの種別に▲を指定している．一方，教員は入力データ数の出力の検査に関しても重み付けを低くしている．

(5) パターン 24 は，main 関数の引数 void の綴りミスの誤りである．gcc コンパイラが引数の誤りを学生に通知していれば学生は修正するため，このような採点誤りは避けられた．

教員の評定法の特徴をまとめると次のようになる．

(1) 全体の構造が正しいかを確認し，その後で細部のコードを確認する．

(2) 出力が正しいかよりもコードの構造が正しいことを重視する．

(3) テストケースの重みを変えている．また，出力メッセージ中の数値の検査も重みを変えている．

(4) 課題の学習項目の理解に関して，出題者の意図と異なる場合がある．

教員の評定値と WC および C とのそれぞれ残差平方和を比べると，それぞれ 217 と 214 で差がない．教員ごとの残差平方和の平均値と標準偏差は，評定値 WC が 20.2 と 31.0 で，評定値 C が 30.6 と 19.5 で教員間の評定値の差が大きく，また，評定値 WC に近い評定をしている教員が多いことがわかる．

5. 評定と進捗の関係の分析

評定値は学生の作成プログラムの品質を表すが，評定値によって学生の進捗を示すことができないかと考え，学習履歴データを分析し，評定値と進捗状況との関係を調べて

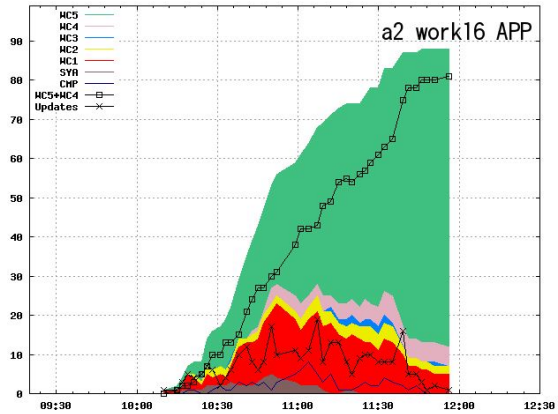


図 5 評定法 WC による課題 6 の進捗グラフ

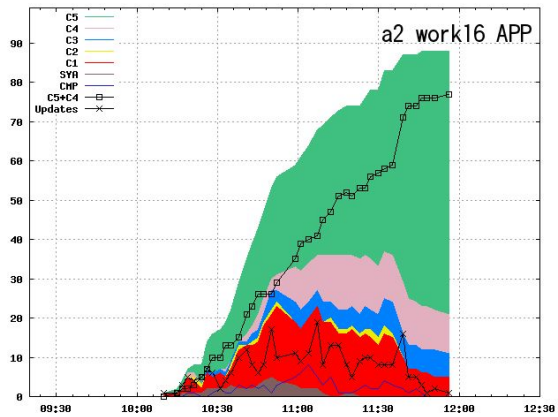


図 6 評定法 C による課題 6 の進捗グラフ

みた．

学生への指導の内容やタイミングは教員によって異なるため，進捗はクラスで異なる．そのため，ここでは a2 クラスの学習履歴データを見ていく．評定法 WC と C による課題 6 の進捗グラフを図 5 と図 6 に示す．図 5 において，WC5〜WC1 は等級 5 から 1 までの各学生数，SYA はベースファイルから修正がない種別「写」の学生数，CMP はコンパイルエラーとなっている学生数，WC5+WC4 は合格レベルである等級 5 と 4 の学生数の合計，Updates は前回の検査以降にファイルを更新した学生数である．積み上げ折れ線図で色を分けて学生数の時間変化を示している。一方，図 6 において，C5〜C1 は等級 5 から 1 までの各学生数で，C5+C4 は合格レベルの等級 5 と 4 の学生数の合計で，他の記号は図 5 と同じである．

12 時前でグラフが終わっているのは 12 時から小テストを実施したためである．合格となった学生数は，a2 クラスの出席学生の 88 名のうち，図 5 では 'WC5+WC4' の線が示す 81 名と図 6 では 'C5+C4' の線が示す 77 名である．2 つの図を比べて目立つ違いは，図 5 では WC2 が多く，図 6 では C4 が多い．

a2 クラスでは，教員が課題 6 の解答例のプログラムを 11 時 25 分に公開したので，11 時 35 分ぐらいから'WC5+WC4'

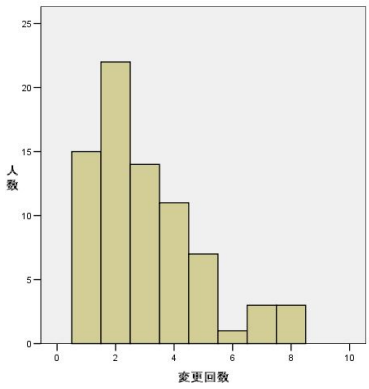


図 7 変更回数の分布

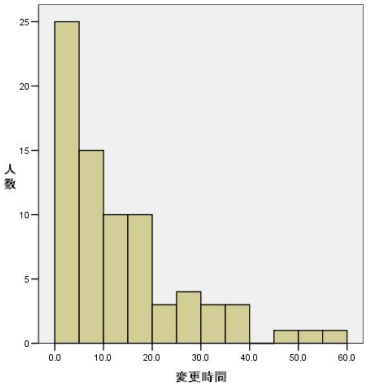


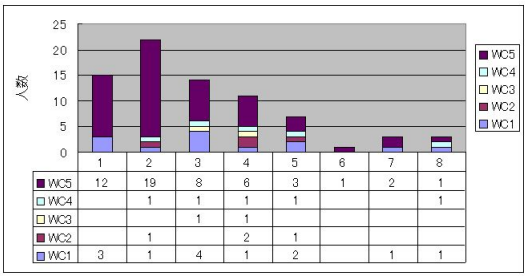
図 8 作業時間の分布

と'C5+C4' の線が急激に上昇している．11 時 25 分以降の評価は学生の理解度とは関係ないので，以降は 11 時 25 分以前のデータに絞って分析する．

11 時 25 分までにプログラムを作成した学生 77 名の変更回数と作業時間を図 7 と図 8 に示す．変更回数は平均値が 1.8，標準偏差が 1.84，最小値が 1 で最大値が 8 である．一方，作業時間は平均値が 12.37，標準偏差が 13.05，最小値が 0 で最大値が 57 である．ただし，作業時間とは最初にファイルを変更した時刻から 11 時 25 分前までの最後に変更した時刻との差 (単位は分) である．そのため，1 回だけファイルを書いた場合は 0 となる．作業時間は学生によって大きくことなるため，以降は変更回数について分析する．

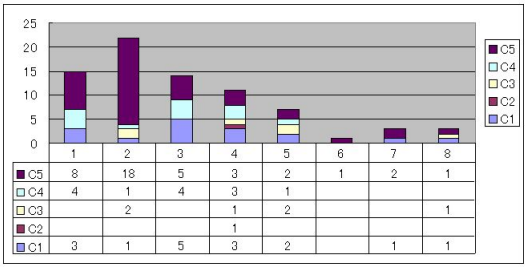
次に，11 時 25 分において 5 つの評定値と判定された学生が一体どのような修正行為をしているかを分析する．図 9 と図 10 は，11 時 25 分の時点のそれぞれの評定値と更新回数のクロス集計である．例えば，図 9 を見ると，1 回の修正で WC5 となった学生は 12 名，2 回は 19 名，3 回は 8 名で課題 6 はこれらの学生にとっては簡単なプログラミング課題であったろう．6 回以上の修正で WC5 となった粘り強い学生も 4 名いる．一方，WC1 と判定された学生も多様である．1 回しか変更していない学生が 3 名，6 回以上変更したが WC1 のままである学生が 2 名いる．

図 11 は各変更回における評定値ごとの学生数を遷移を



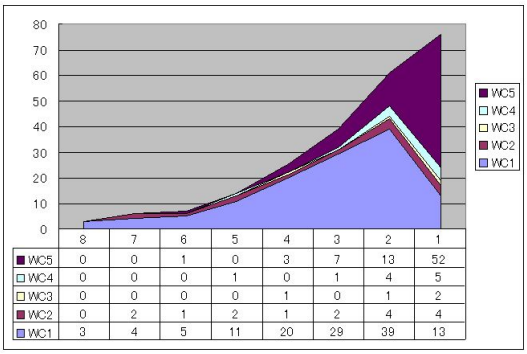
	1	2	3	4	5	6	7	8
WC5	12	19	8	6	3	1	2	1
WC4		1	1	1	1			1
WC3			1	1				
WC2		1		2	1			
WC1	3	1	4	1	2		1	1

図 9 評定法 WC による公開直前の評定値と変更回数の関係



	1	2	3	4	5	6	7	8
C5	8	18	5	3	2	1	2	1
C4	4	1	4	3	1			
C3		2		1	2			1
C2				1				
C1	3	1	5	3	2		1	1

図 10 評定法 C による公開直前の評定値と変更回数の関係



	8	7	6	5	4	3	2	1
WC5	0	0	1	0	3	7	13	52
WC4	0	0	0	1	0	1	4	5
WC3	0	0	0	0	1	0	1	2
WC2	0	2	1	2	1	2	4	4
WC1	3	4	5	11	20	29	39	13

図 11 評定法 WC の評定値の学生数の遷移

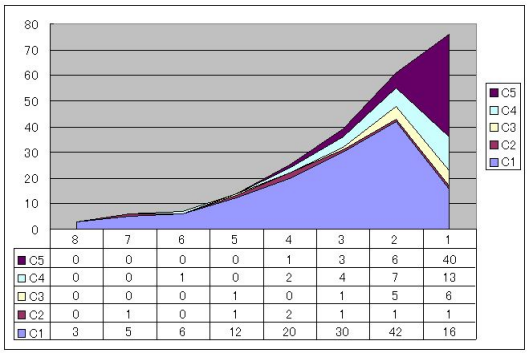
示す．X 軸の 8 から 1 の数値は 11 時 25 分から前に振り返ってみた回数を表す．例えば，WC1 の 8 回目の学生数の 3 は，8 回修正した学生のうち 3 名が WC1 であったことを示す．WC1 の 7 回目の学生数の 4 は，7 回以上修正した学生のうち後ろから数えて 7 回目の修正で 4 名が WC1 であったことを示す．

図 11 と図 12 を比べても評定法による違いはわからないが，変更回と評定値のスピアマンの順位相関係数を調べると，1%の水準で評定法 WC が-0.529 で評定法 C が-0.548 となり，評定法 C の方が相関が高い．

また，進捗度を測る方法として，プログラムの変更に対する敏感さがある．前回と評定値が変わった回数を数えると，評定 WC では平均値が 1.24 で標準偏差が 1.56 ，評定 C では平均値が 1.17 で標準偏差が 1.60 となり，評定 WC の方が若干感度が良い．

6. 評定法の教員アンケート調査

4 章で述べた教員による作成プログラムの評価といっしょに，評定法に関するアンケート調査をおこなった．そ



	8	7	6	5	4	3	2	1
C5	0	0	0	0	1	3	6	40
C4	0	0	1	0	2	4	7	13
C3	0	0	0	1	0	1	5	6
C2	0	1	0	1	2	1	1	1
C1	3	5	6	12	20	30	42	16

図 12 評定法 C の評定値の学生数の遷移

課題	時刻	変更	5	4	3	2	1	写	★	CMP
work11	11:19	1	82	2	1	3	1	1	10	0
work12	11:48	1	75	0	2	2	6	0	15	2
work13	11:50	1	81	6	2	0	1	0	10	1
work14	11:57	1	81	3	2	1	3	0	10	2
work15	11:22	1	75	10	1	1	0	2	11	0
work16	11:56	1	76	5	2	2	3	0	12	0
work17	11:55	5	77	1	2	0	8	0	12	2
work18	11:57	3	58	2	5	1	10	0	24	4
work19	11:59	27	38	0	3	0	26	1	32	5
work1a	11:58	1	21	0	1	1	12	1	64	5

図 13 a2 クラスの課題ごとの進捗表示

の結果を表 4 に示す．質問の 1 から 7 までが評定法に関する質問，質問の 8 から 13 までが評定法の利用に関する質問，質問の 14 が検査項目に関する質問である．

結果で意見が分かれたのは質問 5 と 6 の評定法 WC と評定法 C のどちらが良いかということであるが，どちらかと言えば評定法 WC の方が妥当であるという判断になった．また，質問 7 の等級 1 から 3 までを判定する点数の範囲も意見が分かれた．

Hercules システムは，提案した評定法で評価した結果を用いて，図 5 の課題の進捗グラフのほかに，クラスの課題の進捗状況の図 13 のようにブラウザに表示する．同様に，全クラスの進捗状況や各学生の進捗状況をブラウザに表示する．これらの評定法の利用については全体的に有効であるという評価である．図 5 や図 13 を見てクラスの進捗状況が把握できるかという質問 8 に関しては不十分と考えている教員がいる．

7. おわりに

本稿では，プログラミング課題の仕様を定義し，それに基づいた評定法として，パスした検査項目の個数をその種別で重み付けして点数化し，評定値を判定する方法 (評定法 WC) を提案した．配列の参照・代入を学習項目とする最大値を計算するプログラミング課題を例にとって，27 種類の学生の誤りパターンに対する評定値を教員による自由

表 4 評定法の教員アンケートの結果

	質問内容	はい	どちらでもない	いいえ
1	表 1 の課題の仕様を 4 つの種類にわける基準は明確ですか？	5	1	0
2	表 2 の等級 5、4 の定義は学生のプログラムの評価として妥当ですか？	5	1	0
3	表 2 の等級 3、2、1 の定義は学生のプログラムの評価として妥当ですか？	4	1	1
4	表 3 のように、等級 5 と 4 に学習項目と入出力仕様の誤りがない条件を入れるのは妥当ですか？	6	0	0
5	点数の計算法として、誤りの個数に検査の種別ごとの重み付けをするのは妥当ですか？	4	2	0
6	点数の計算法として、種別ごとの重み付けせずに誤りの個数で計算する方が妥当ですか？	1	3	2
7	表 3 のように、等級 3，2，1 を判断するための点数の範囲 (例えば 70 点以上) は妥当ですか？	3	3	0
8	本評定法によって、クラスの進捗を正確に判断できるようになりましたか？	4	2	0
9	本評定法によって、他のクラスとの進捗を比較することが容易になりましたか？	5	1	0
10	本評定法によって、進捗の悪い課題を把握するのが容易になりましたか？	6	0	0
11	本評定法によって、進捗の遅い学生を見つけるのが容易になりましたか？	4	1	1
12	本評定法によって、学生へのフィードバックをするタイミングが明確になりましたか？	3	1	2
13	本評定法によって、学生へのフィードバックをする内容が明確になりましたか？	2	1	3
14	表 1 のように、C 演習 II の採点スクリプトは 4 つの仕様に対応した種別が指定されていますか？	5	1	0

な評定値と比べた．その結果，教員の評定値に大きな差があり，また，1) 全体の構造が正しいかを確認し，その後で細部のコードを確認する，2) 出力が正しいかよりもコードの構造が正しいことを重視する，3) テストケースの重みや出力メッセージ中の数値の重みを変えている，など本システムの検査法とは異なることがわかった．

また，学習履歴データを使って，進捗との関係を調べた．特に，パスした検査項目の個数を点数化し評定値を計算する方法 (評定法 C) と比較した．変更回数と評定値の遷移との相関や変化の回数や，グラフで比較しても明確に区別できるほど差がなかった．

最後に，教員アンケートによって評定法の妥当性と利用性を調査した．評定法 WC と評定法 C を比べたが，評定法 WC と評定法 C のどちらが妥当かを質問したが意見が分かれた．一方，本評定法を用いてクラスの進捗，課題の進捗，学生の進捗のグラフや表で表示するのは有用性が高いと評価された．

本稿では，20 行の小さなプログラムを例に分析したが，もっと大きなプログラムや関数が幾つもある複雑なプログラムに対しても分析をし，評定法の妥当性を評価したい．また，教員の評定法のよい点を取り入れるなどして本システムの評価法を改善していきたい．

謝辞 多忙の中，プログラムの評定結果の調査にご協力頂いた，大阪工業大学情報科学部の C 演習 II を担当している教員各位にに深く感謝する．

参考文献

[1] Ala-Mutka, K.: A Survey of Automated Assessment Approaches for Programming Assignments, *Computer Science Education*, Vol. 15, No. 2, pp. 83–102 (2005).
[2] Douce, C., Livingstone, D. and Orwell, J.: Automatic test-based assessment of programming: A review, *Educational Resources in Computing*, Vol. 5, No. 3 (2005).
[3] 内藤広志，斉藤　隆：プログラミング演習のための進捗モニタリングシステム，情報処理学会研究報告 コンピュータと教育研究会報告， Vol. 2008, No. 13, pp. 33–40 (2008).
[4] 内藤広志，斉藤　隆，水谷泰治：GUI プログラミング課題の自動採点方式について，情報処理学会研究報告 ソフトウェア工学研究会報告， Vol. 2008, No. 93, pp. 81–88 (2008).
[5] 内藤広志：演習形式による「構造化文書処理」の実施と問題点，VMA 研究会， Vol. 24, pp. 7–15 (2009).
[6] Olson, D. M.: The reliability of analytic and holistic methods in rating students' computer programs, *9th SIGCSE technical symposium on Computer science education*, pp. 293–298 (1988).
[7] Smith, L. and Cordova, J.: Weighted primary trait analysis for computer program evaluation, *Computer Science Education*, Vol. 20, No. 6, pp. 14–19 (2005).
[8] Joy, M., Griffiths, N. and Boyatt, R.: The BOSS online submission and assessment system, *Educational Resources in Computing*, Vol. 5, No. 3 (2005).