

小コンテスト形式の 初級 C 演習における教師支援 - 解答状況の時系列表示による モニタリング機能の試用実験 -

西村 智治[†] 川崎 慎一郎[†] 富永 浩之[†]

大学情報系での初級 C 授業に競争型学習を取り入れ、小コンテスト形式でのプログラミング演習を提案している。中間目標として、部分仕様に基づく実行テスト系列を用意し、段階的実装を誘導する。大会支援サーバ tProgrEss では、提出されたソースコードの正誤を自動判定する。また、進捗状況を即時通知し、順位公開して、競争意欲を刺激する。教師支援として、解答状況のモニタリング機能を開発した。試用実験を行い、提出履歴から教育効果を確認し、アンケート結果から UI と機能を改良した。

Teacher Support System in Introductory C Programming Exercises Based on Competitive Learning - An experiment of Visualization Pages of Answering Situation in Monitoring Module -

Tomoharu Nishimura[†] Shinichiro Kawasaki[†]
Hiroyuki Tominaga[†]

To improve introductory C language lesson in an engineering college, we have proposed a small contest style exercise in classroom. We have developed tProgrEss, the contest management Web server. It judges an uploaded program by an execution test with input and output data. To adjust a contest for beginners, we offer several preparation tests, which are for step-by-step sub-goals as partial specification. We set the rule of partial points and plus or minus points based on time. The system shows the result of judgment and exhibits the ranking page immediately. We are developing monitoring functions as the teacher support sub-system. It offers time-series view of answering situation. We carried out an experiment and questionnaires. We improved the UI and functions.

1. はじめに

大学の情報系学科では、初年次の C 言語教育を重視している。しかし、理解度が異なる多様な学生が受講しており、進捗状況や演習態度に大きな差が生じやすい[1]。本研究では、授業中の演習を活性化させるため、初心者が気軽に参加できる小コンテスト形式の演習を提案している[2][3]。1回のコンテストで数問のプログラミング問題を提示する(図 1)。ローカル PC で解答コードを作成し、大会 Web サーバ tProgrEss にアップロードする[4]。サーバ側でコンパイルした実行バイナリに、入力サンプルを与えて実行し、実行結果と出力サンプルを照合して、正誤を判定する(図 2)。ただし、初心者がプログラムを完成するには時間がかかり、途中で諦めてしまう可能性がある。そこで、一部の仕様を満たす解答に対しても部分点を与えるように、実行テスト系列を導入する[5]。また、各学生の進捗状況を即時的に公開し、競争意欲を高める。また、コンテスト中に教師の指導を支援するモニタリング機能も試作している。

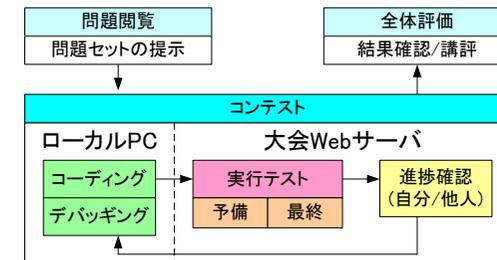


図 1 コンテスト形式の演習の進行

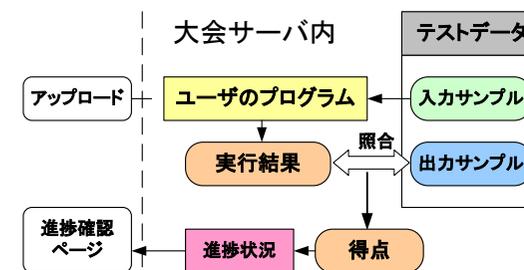


図 2 解答コードの判定手順

[†] 香川大学工学部
Faculty of Engineering, Kagawa University

2. 演習形式と運営サーバtProgrEssの概要

2.1 コンテスト形式の演習の実施形態

コンテストの実施形態は、教室型/宿題型/試験型を用意する。これらは、異なる機能を提供するのではなく、コンテストの要項や問題編成の指針を示すテンプレートである(図3)。教室型は、毎回の授業の最後に、総まとめとして、教室内で一斉に実施する。基本的な事項で、例題の類題として、15~30分で解ける問題を出題する。宿題型は、1~2週間の期間内とし、学外からのアクセスも許容する。応用的な事項の組合せで、60~120分の問題を出題する。試験型は、成績に関与する演習で利用する。教室型より長い、制限時間を厳格にし、難易度の高い問題を出題する。

2.2 実行テスト系列による正誤判定

コーディングの進め方を誘導したり、プログラムの作成指針を示すため、複数の予備テストと1つの最終テストからなる実行テスト系列を採用する(図4)。予備テストは、3つ程度の中間目標として構成し、仕様を緩めて徐々に完成に近づけさせる(図5)。予備テストは、入力制限と出力許容に分けられる[6]。入力制限では、仕様で想定される範囲内の一部のデータでのみ正しく実行できればよく、例外処理を必要としたり、継続条件や分岐条件で境界値となるものを避ける。出力許容では、仕様で要求される出力の書式において、照合基準を緩和し、データ列の平均と最大値を求める問題で、平均だけの出力が合えばよいなどとする(図6)。

2.3 得点ルールと誘導

プログラムの判定結果には、時間調整点と誤答減点を導入した得点ルールを適用する。時間調整点は、提出時期によって、通常の配点に加減点を付与する(図7)。事前期間の提出は、コンテストの開始前なので無効とし、得点を0にする。早期期間の提出は、積極性を評価して加点する。通常期間の提出には、加減点はない。延長期間の提出は、多少は許容する代わりに減点する。事後提出は、追試的な措置であり、参加点のみとする。無効提出は、授業終了後にシステムの正誤判定の機能のみを使用する際の提出であり、得点は0にする。早期と延長における加減点は、提出時期で傾斜させる。これらを設けることで、学生の早期解答を促し、演習の活性化を狙う。一方、誤答減点は、最終テストの誤答に減点のペナルティを与える。最終テストでは、例外的な入力を含む、網羅的な入出力サンプルを用意する。誤答減点により、完答への慎重な確認を行わせる。最終テストが誤答や未答で終わった場合は、予備テストの部分点のみを与え、誤答減点を行わない。

2.4 大会運営サーバtProgrEssの機能

上記のような演習を実施するため、大会運営サーバtProgrEssを開発し、実際の授業で運用している[7][8]。実行テストの結果は、図8の6段階で判定する。判定結果、およびコンパイル時のメッセージと実行結果は、判定確認ページで学生に提示する(図

9)。学生は、これらの情報を基に、解答の修正や、次のテストに取り組む(図10)。また、順位表示ページで、各自の判定結果は教室全体にも公開される。他人の進捗状況を見て、競争意識を高め、意欲を持続させる。

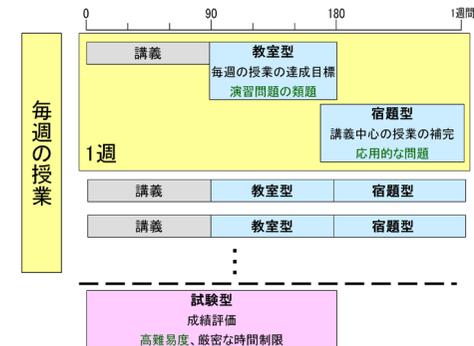


図3 演習形態に応じた3種類のコンテスト型

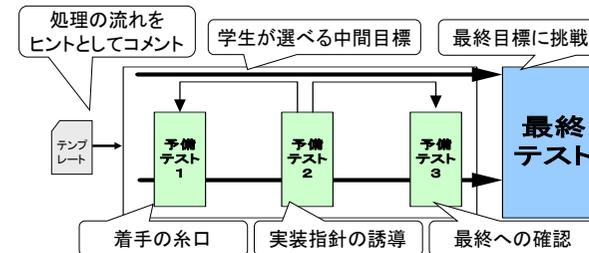


図4 実行テスト系列の予備テストと最終テスト

予備テスト1: 入力値をデータとして正しく格納	
入力の順序とデータ型	不適な入力範囲の確認
不定個入力の番兵での打ち切り	入力値の計数と振分け
予備テスト2: 主要な反復処理や配列操作	
反復の構造(パス、ネスト)	配列のシフト(追加・削除・逆順)
反復の範囲(初期値、継続条件)	配列の加工(分割・併合・生成)
予備テスト3: 整形出力や例外処理	
整形出力(桁数、区切、表形式)	例外処理(レアケース)
	境界値検査

図5 実行テスト系列における確認項目

不定個の整数値を入力し、平均値を出力する。
負数を番兵とし、小数点以下2桁を出力せよ。

番兵式による入力の確認	予備テスト1
5 -1	5.000000
平均値計算の確認	予備テスト2
2 3 4 5 -1	3.500000
出力書式の確認	予備テスト3
1 2 3 4 5 6 -1	3.50
十分な量の入出力サンプルを用い、プログラムの正誤を判定する。	最終テスト

図 6 入出力サンプル例

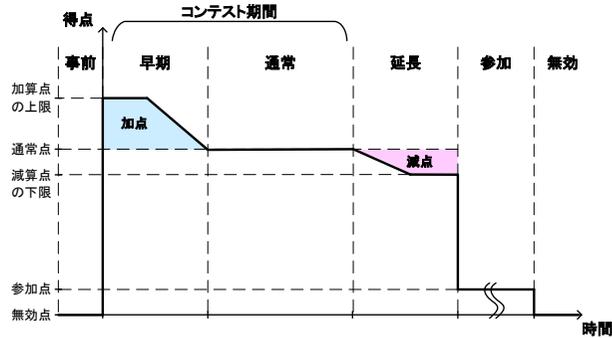


図 7 時間による得点の変化

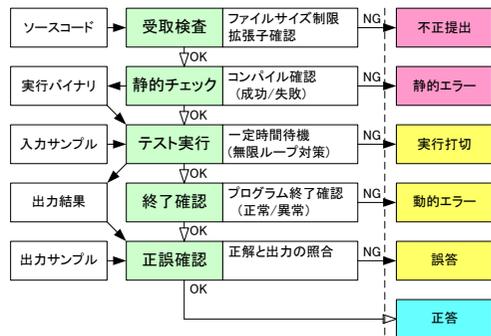


図 8 実行テストの手順

予備テスト進捗

問題	1	2	3	1	2	3	Q04	Q05	Q06	Q0	総合得点	総得点	
予備	○	○	○	○	○	○	△	○	○	○	△	○	590
最終	100	60	100	60	70	100	0	100				1	

最終テスト進捗

個人 教室 テンプレート

ランキング

テスト	区分	得点	概要	入力	出力
予備01	入力限定	40 %	計算結果が整数値になる場合のみに対応させる。	10.0 5.0	25.000000
最終		100 %	問題文の通り		

解答ソースコードの提出

● 実行テスト T01 40点 入力限定
計算結果が整数値になる場合のみに対応させる。

テスト情報の切換え

図 9 問題閲覧ページの出題解答モード

テスト	区分	得点	概要	入力	出力
予備01	入力限定	40 %	計算結果が整数値になる場合のみに対応させる。	10.0 5.0	25.000000
最終		100 %	問題文の通り		

照合結果

コンパイル結果

実行結果

入力サンプル	出力サンプル	実行結果
10.0 5.0	25.000000	25.000000

提出コード

```

//=====
// 工学部09年度「プログラミング1」(期末) 宿題 09-01
// 2009.10.28(月) 09:287 川崎慎一郎
//=====
    
```

図 10 問題閲覧ページの採点結果モード

3. 教師支援としてのモニタリング機能

3.1 モニタリング機能の要件

学生側の支援だけでなく、教師側への支援も検討している。まず、学生個人に提示する情報を集約し、学生全体を総覧できるページを用意する。その上で、全体への通知や個人への助言など、即時的な指導への支援の強化を図る。各学生の問題への着手状況を一覧するだけでなく、指導側が注意を払うべき部分を発見しやすくする機能が必要である。そこで、教員の指導を補助するモニタリング機能を追加する。例えば、提出が滞っている学生を自動通知する。ポップアップや色分けなどで、教員による監視を支援する。進捗状況の検出には、システムログを利用する。ユーザの行為を時系列で分析して、演習態度の注意やヒントの提示を行う。実行結果の正誤判定だけでなく、教師の目視によるコード確認も行い、各学生への個別指導を支援する機能も導入する。これらを踏まえ、運用実験を行い、教育効果を検証する。

3.2 モニタリング機能の概要

モニタリング機能として、学生全体の進捗状況を一目で把握できるようにして適切な助言や指導が行えるようにする得点順位ページ、提出された解答コードを閲覧して学生へ適切な指導が行えるようにするための提出履歴ページ、各学生の提出状況を把握して提出していない学生や問題に詰まっている学生を発見するための全学生と学生別の提出状況ページ、各問題の着手人数を把握して問題の難易度の適切さを判断するための学生単位の着手集計ページ、各問題の提出数を把握してどの問題が解かれているか問題が適切か判断するための解答単位の着手集計ページを作成する。

3.3 モニタリング機能の一覧表示ページの機能

得点順位ページは、参加学生の総得点による順位表示を行う。各問題の実行テストごとの正誤判定の結果と得点も表示する(図 11)。コンテスト中に、学生全体の進捗状況を把握するために用いる。現時点では、学生側のページとほぼ同じである。ただし、学生や答案に対して、他の教師側ページへのリンクを張っている。

提出履歴ページは、提出された解答コードを時系列で表示する。6段階の判定結果を示し、解答コードと実行結果へのリンクを張る(図 12)。コンテスト中に、学生が提出した解答コードを閲覧する。各項目の指定を行うことで、絞り込み表示が行える。

3.4 モニタリング機能の詳細分析ページの機能

全学生の提出状況ページでは、全学生の実行テストごとの提出回数を一覧する(図 13)。学生を選ぶと、学生別の提出状況ページに移動する(図 14)。いつどの実行テストに取り組んだかが時系列で表示される。コンテストの最中と事後に、特定の学生の振舞いを把握するために用いる。

学生単位の着手集計ページでは、実行テストごとの着手人数と着手率、および正答者数と正答率を表示する(図 15)。下限未満の正答率の箇所に色付けを行う。コンテ

スト中に、未着手のままの問題を発見するために用いる。このような問題に対して、ヒントなどを全体に通知する。

解答単位の着手集計ページでは、実行テストごとの着手件数と着手比率と、正答件数と正答率を表示する(図 16)。下限未満の正答率の箇所に色付けを行う。提出件数が多く、正答率が低い実行テストは、適切でない出題である可能性がある。必要があれば、訂正や補足、採点時の配慮を全体に通知する。また、問題ごとに、各実行テストの時間軸に対する提出件数のグラフを表示する。

Lesson : RiseProg1 2010 Contest : W13 H01 更新

	Q01			Q02			Q03			Q04			Q05			Q06			Q07			Q08			順位
	予1	予2	予3	予1	予2	予3	予1	予2	予3	予1	予2	予3	予1	予2	予3	予1	予2	予3	予1	予2	予3	予1	予2	予3	
s10t263	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	1
	CX	180	CX	CX	165	CX	180	CX	180	CX	180	CX	220	CX	1545										
s08t207	△	△	△	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	2
	CX	180	CX	180	CX	175	CX	170	CX	170	CX	215	CX	220	1530										
s10t231	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	3
	CX	180	CX	160	CX	180	CX	180	CX	170	CX	220	1530												
s09t234	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	4
	CX	170	CX	170	CX	170	CX	170	CX	170	CX	200	CX	220	1490										

図 11 得点順位ページ

コンテスト 問題 テスト 結果 学籍番号 表示数
 最新100

提出	日時	時差	コンテスト	ユーザ	問題	テスト	判定	解答	実行	割符
36881	2011-01-27 20:51:35	-----	W13_H01	s10t232	Q06	最終	誤答	解答	結果	検出
36880	2011-01-27 20:49:57	00:01:38	W13_H01	s10t226	Q01	予備02	正答	解答	結果	検出
36879	2011-01-27 20:48:11	00:01:46	W13_H01	s10t232	Q06	最終	誤答	解答	結果	検出
36878	2011-01-27 20:47:13	00:00:58	W13_H01	s10t232	Q06	予備03	正答	解答	結果	検出
36877	2011-01-27 20:46:06	00:01:07	W13_H01	s10t203	Q02	最終	誤答	解答	結果	検出
36876	2011-01-27 20:45:53	00:00:13	W13_H01	s10t203	Q02	予備02	誤答	解答	結果	検出

図 12 提出履歴ページ

	Q01			Q02			Q03			Q04			Q05			Q06			Q07			Q08			得点			
	予1	予2	予3		最終提出																							
s06t201	---	1	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	170
	01:27	16:41	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	01:27 16:41
s06t234	29	7	---	3	1	1	1	1	1	2	1	1	1	1	1	1	1	2	2	1	1	1	1	1	1	1435		
	01:31	19:37	---	01:27	17:17	---	01:27	04:26	---	01:27	17:25	---	01:27	03:55	---	01:27	03:40	---	01:27	03:26	---	01:27	03:11	---	---	01:31 19:37		
s07t201	2	1	---	1	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	170
	01:27	00:16	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	01:27 00:16
s07t245	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	0
	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

図 13 全学生の提出状況ページ

経過時間 (時間)	Q01		Q02		Q03		Q04		Q05		Q06		Q07		Q08	
	予1	予2	予3	最終												
0 - 12	1	1	1	1	1	1	1	1	1	1	2					
1	1	1	1	1	1	1	1	1	1	1	2					
2																
3																
4																
5																
6																
7																
8																
9																
10																
11																
12 - 24																
24 - 36																

図 14 学生別の提出状況ページ

受講者人数 93人
 下限正答率 70% 未滿

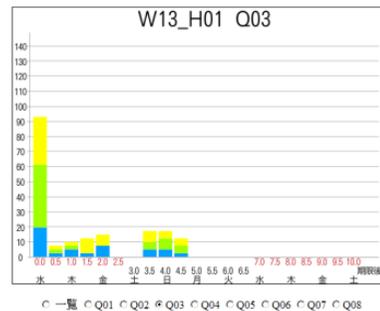
番号	予備1	予備2	予備3	最終
[a001] 着手	29 31%	23 25%	-	36 39%
[a001] 正答	24 83%	22 96%	-	34 94%
[a002] 着手	19 20%	22 24%	-	23 25%
[a002] 正答	18 95%	17 77%	-	17 74%
[a003] 着手	15 16%	16 17%	-	17 18%
[a003] 正答	15 100%	15 84%	-	16 84%
[a004] 着手	8 9%	8 9%	-	11 12%
[a004] 正答	8 100%	8 100%	-	11 100%
[a005] 着手	9 10%	8 9%	-	10 11%
[a005] 正答	7 78%	7 88%	-	9 90%
[a006] 着手	4 4%	3 3%	6 6%	6 6%
[a006] 正答	4 100%	3 100%	5 83%	6 100%
[a007] 着手	3 3%	3 3%	2 2%	5 5%
[a007] 正答	3 100%	3 100%	2 100%	5 100%
[a008] 着手	3 3%	3 3%	3 3%	5 5%
[a008] 正答	2 67%	3 100%	2 67%	5 100%

図 15 学生単位の着手集計ページ

受講者人数 93人
 下限正答率 20% 未滿

番号	予備1	予備2	予備3	最終
[Q01] 着手	51 55%	41 44%	-	60 65%
[Q01] 正答	43 84%	39 95%	-	57 95%
[Q02] 着手	40 43%	42 45%	-	47 51%
[Q02] 正答	38 95%	37 88%	-	35 74%
[Q03] 着手	32 34%	35 38%	-	42 45%
[Q03] 正答	31 97%	34 97%	-	40 95%
[Q04] 着手	20 22%	18 19%	-	27 29%
[Q04] 正答	19 95%	18 100%	-	26 96%
[Q05] 着手	25 27%	24 26%	-	30 32%
[Q05] 正答	23 92%	22 92%	-	29 97%
[Q06] 着手	8 9%	9 10%	12 13%	14 15%
[Q06] 正答	8 100%	5 56%	10 83%	14 100%
[Q07] 着手	9 10%	9 10%	6 6%	13 14%
[Q07] 正答	9 100%	8 89%	4 67%	12 92%
[Q08] 着手	8 9%	7 8%	6 6%	11 12%
[Q08] 正答	5 63%	5 71%	2 33%	10 91%

図 16 解答単位の着手集計ページ



4. 教育実践での運用評価

4.1 初級C演習授業における運用実践の概要

システムの評価として、実際の授業で運用した。対象科目は、学科1年次後期の初級C演習「プログラミングI」である。受験者は、受講者ほぼ全員の80名である。コンテストは、宿題型として実施した。出題範囲は、変数とデータ型、演算と式、選択構文と反復構文、配列、関数である。

コンテスト期間を1週間とし、8問を出題した。基本問題を5問、応用問題を3問とした。基本問題は、配点150点、予備テスト2個(50/100点)である。60分程度で解答できる基礎的な問題とする。応用問題は、配点200点、予備テスト3個(50/100/150点)である。90分程度で解答できる応用的な問題とする。合格基準としての目標得点を750点とし、完答3問よりも半答6問を目指すように指導した。時間調整点については、図7による説明だけでなく、表1を明記して周知した。

4.2 全体での解答コードの提出履歴の時系列分析

実験結果として、解答コードの提出履歴を分析した。図17は、解答コードの提出状況を時系列でグラフ化している。初日の提出が1500件中の350件で、全体の5分の1以上を占めている。早期提出による加点を狙って、すぐに解答に着手している。早期と通常で、期間の終了前12時間に、提出の山がある。時間調整点の導入により、以前より、前倒しで解答に取り組む傾向が出ている[7][8]。通常期間の土～日は正答率が0.7~0.8(50~100件)で、各締切前や遅延提出の0.4程度(100~150件)より高い。この期間は、確実な正答を目指している。一方、早期と通常の期間の終了前は、正答率が0.4程度(100~150件)で、各期間中の0.5~0.8(50~350件)より低い。締切を意識し、慌てて正誤を確認している傾向もみられる。遅延期間の締切前の正答率は0.6(100件)で、それ以前の締切前の0.4程度(100~150件)より高い。短期間での連続した提出などから、他人の解答コードを丸移した可能性がある。実際に、そのようなケースも見つかった。また、何度も誤答を繰り返している問題に対して、該当する解答コードを集中して確認し、頻出する間違いを指摘できた。例えば、変数の初期化をせずに参照するなど、コンテスト中にヒントを追加した場合もある。

4.3 成績別での解答コードの提出履歴の時系列分析

さらに、解答コードの提出履歴を成績で4群に分けて分析した。図18は、それぞれの集団ごとに、解答コードの提出状況を時系列でグラフ化している。第1群は、提出数が最も多く約530件である。全体の提出割合は、加点期間で7割、通常期間が2.5割、減点期間が0.5割であった。時間調整点による加点への意識が高い。実行テストの利用では、予備1、2、3をバランスよく使い、最終で誤答しないようにしている。第2群は、提出数が第1群に次いで多く約450件である。全体の提出割合は、加点期間が6割、通常期間が2割、減点期間が1割であった。時間調整点の加点と減点を意

識している。実行テストの利用では、最終の提出数が多く、誤答率も高くなっている。最終への慎重な解答を促す必要がある。第3群は、提出数が第2群より少なく約380件である。全体の提出割合は、加点期間が4割、通常期間が3割、減点期間が3割であった。加点期間がやや高いが、時間調整点への意識はあまり高くない。実行テストの利用では、予備1、2をメインに提出しており、最終の正答率も高くなっている。これは、提出件数が上位群と比べ少ないことから、難しい問題で躓き、そのまま提出を諦めたためと考えられる。問題で行き詰っている学生への助言が必要である。第4群は、提出数が最も少なく約160件である。ほとんど提出がなく、コンテストに参加していない。問題への着手を促す必要がある。

上位群は、実行テストや時間調整点を上手く活用し、目標点を取れている。今後は、システムが判断できない、プログラムの実装方法やコードの書法などの、よりレベルの高い指導が必要である。そのためには、実装方法が非効率的な学生の検出など、モニタリング機能を強化する必要がある。また、学生への個別指導と全体通知などができるチュータリング機能を検討する。下位群は、全体的に提出数が少ない。原因として、解けない問題で詰まり解答を諦めた、締切前に解答を始め時間がなくなった、提出自体していないなどが考えられる。そのため、そのような学生をシステムが自動的に教師に通知するなど、モニタリング機能の強化が必要である。また、コンテスト中に学生の振舞いから、システムが自動的に問題の助言、解答を促す通知などができるチュータリング機能を検討する。

4.4 アンケート分析

受験者には無記名で、4段階の客観式アンケートを実施した。自己申告の得点によって、上位から下位まで4群に分けて集計した(表2)。システムのユーザビリティについては、応答速度と順位表示は、評価が高かった。しかし、順位表示では、第1群と第4群から、学籍番号ではなく匿名で表示してほしいという意見があった。これは、第1群はできていない学生から教えてほしいと請われることが多く、第4群は他の学生にできていないと知られることを嫌がったためと考えられる。時間調整点の効果については、上位群は意識して早めに提出していたが、最下位群では利用への意識が十分とは言えなかった。彼らを問題着手へと誘導するための支援が必要である。

4.5 教師支援機能の評価

授業の担当教員に対し、コンテスト中の教師側ページの利用について、インタビュー形式のアンケートを実施した[9]。コンテスト中の利用について、問題に取り組んでいない学生の検出、誤答に対する解答コード上の間違いの発見、などが挙げられた。UIの評価では、機能ごとにページが分かれているが、1つのページで多くの機能が使用できると良い、との意見であった。教師支援機能については、現状の5つのページから、学生監視用と問題監視用の2つのページを新たに設置し、機能を集約することを検討する。学生側機能についての意見もあり、ランキングにおいて学籍番号を表示す

ると、余り問題が解けない学生が他の学生の目を意識して提出をためらうのではないかと、とのことだった。対策として、匿名表示の検討を進める。

表1 運用実験の宿題型コンテストの得点ルール

得点	1.26(水)	1.27(木)	1.28(金)	1.29(土)	1.30(日)	1.31(月)	2.01(火)	2.02(水)	締切後
難易度B	180	170	160	150	150	150	145	140	15
難易度C	230	220	210	200	200	200	195	190	20

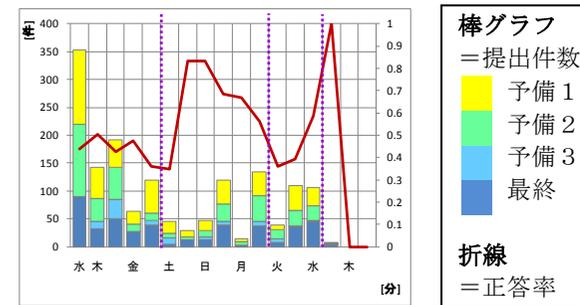


図17 解答コードの提出履歴の時系列分析

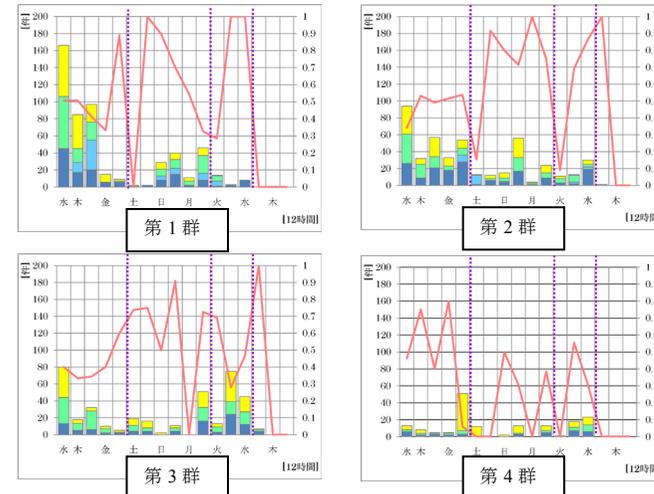


図18 成績別での解答コードの提出履歴の時系列分析

表 2 成績による 4 段階の学生の分類

集団	順位	得点	人数
第 1 群	最上位	900-1600	15 名
第 2 群	中上位	750- 899	21 名
第 3 群	中下位	350- 749	21 名
第 4 群	最下位	000- 349	19 名

5. モニタリング機能のUIの改良

5.1 改良の概要

演習中に教師が、学生の進捗状況や問題の着手状況を把握し易くするため、モニタリング機能の UI を改良した。試用実験から得られた UI 評価の意見を参考に行った。注目すべき意見として、機能別にページがあるためページ遷移が多くなり、どのページを見ているのか把握しにくいというものがあった。これは、教師側ページの数が多く、階層構造に整理されていないためだと考えられる。そのため、まず、教師側ページの利用手順を整理した(図 19)。教師側ページの利用は、始めに、ログインページでユーザ ID、パスワード、担当授業の入力を行う。次に、授業管理ページで、担当授業での新規コンテンツの作成、既存コンテンツの実施監視、コンテンツ全体の成績閲覧を行う。今回は、既存コンテンツの実施監視に対応する、モニタリング機能の 6 つのページについてページ集約を行った。

5.2 教師支援機能のページ集約

モニタリング機能には、学生個人、教室全体、設問の監視対象がある(図 20)。まず、監視対象について整理する。教室全体は、学生全体と設問全体に分けることができる。そのため、監視対象を学生状況と設問状況の 2 つに整理した。そして、それぞれのページについて、学生状況として、得点順位ページ、全学生と学生別の提出状況ページ、設問状況として、解答単位と学生単位の着手集計ページ、提出履歴ページを集約した。

5.3 実施監視ページ

実施監視ページは、授業管理ページで選択された既存コンテンツの状況を総覧する(図 21)。上部には、学生や設問の注目すべき状況を検出順で表示している。横のボタンを選択すると、関連する状況の一覧表示に切り替わる。下部には、学生状況と設問状況ページへのリンクを持つ。コンテンツの最中や事後に、注目すべき状況がないか確認するために用いる。

5.4 学生状況ページ

学生状況ページは、得点順位ページを基にして、全学生と学生別の提出状況ページを集約した(図 22)。切替ボタンを押すことで、テーブルの内容を順位表示と提出状況

に交互に変更する。また、学籍番号を選択すると学生別に詳細な提出状況を表示する。コンテンツ中は、学生に関する情報を一覧し、全体の進捗状況を把握するために用いる。コンテンツ後は、学生の振舞いを把握するために用いる。

5.5 設問状況ページ

設問状況ページは、解答単位の着手集計ページを基にして、学生単位の着手集計ページと提出履歴ページを集約した(図 23)。切替ボタンを押すことで、テーブルの内容を解答単位と学生単位の着手集計に交互に変更する。問題番号には、対応するコンテンツの問題へのリンクが張られている。また、ページ下部には、対応するコンテンツの提出履歴が表示される。コンテンツ中に、問題に関する情報を一覧し、未着手の問題や、適切でない出題を発見するために用いる。

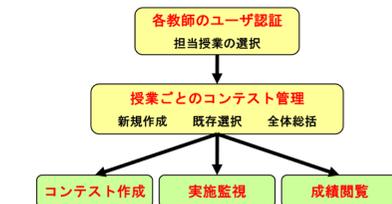


図 19 教師側ページの利用手順

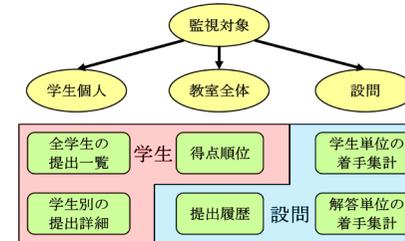


図 20 実施監視ページの監視対象

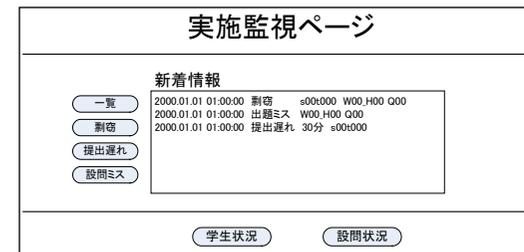


図 21 実施監視ページ

ID	Q01	Q02	Q03	Q04	Q05	Q06	Q07	Q08	順位	得点
10001	○	○	○	○	○	○	○	○	1	1545
10002	○	○	○	○	○	○	○	○	2	1535
10003	○	○	○	○	○	○	○	○	3	1530

図 22 学生状況ページ

提出履歴表示	日時	時差	コンテスト	ユーザー	問題	テスト	結果	学籍番号	表示数
提出	2011-02-02 15:28:08	-----	M3,1401	s100227	Q006	手動00	正答	解法	提出
提出	2011-02-02 14:16:33	01:11:35	M3,1401	s100238	Q005	手動00	正答	解法	提出

図 23 設問状況ページ

6. おわりに

大学情報系学科の入門的 C 授業において、初心者向けの小コンテスト形式でのプログラミング演習を提案し、大会運営の支援サーバ tProgrEss を開発している。これまでの教育実践における運用実験から、教師支援機能の要件を検討し、モニタリングの機能を整理した。基本的なモニタリング機能として、学生の得点順位ページと、解答の提出履歴ページを用意した。詳細な情報提示として、学生別の提出状況の時系列表示や、問題ごとの着手数と正答率の時間的推移を描いたグラフ表示を実現した。

システムの試用実験として、1年次の初級 C の演習科目で、宿題型のコンテストを実施した。時間調整点を設け、提出状況を分析した。解答コードの時系列表示から、

時間調整点が提出を促進する効果が確認できた。また、頻出する誤答から共通の躓きへのヒントを提示したり、不自然な連続提出から剽窃行為を検出するなど、モニタリング機能が有効に利用できた。アンケート回答からは、順位表示における匿名化、教師支援機能のページ集約などの意見が得られた。得られた意見から、教師側ページをログインページと授業管理ページに整理した。また、モニタリング機能の UI の改良として、機能ごとのページを整理し、学生状況ページと設問状況ページに集約した。

今後の課題として、得点順位や提出状況において、特に注目すべき部分を自動的に提示する機能を実装する。チュータリング機能としては、小コンテスト形式の演習に適合したチャットシステムを追加する。例えば、学生全体や学生個別への一斉通知や助言、システムから特定の振舞いを示した学生への通知などの機能が考えられる。指導コメントのテンプレート化や特定の振舞いグループに対する指導などで、教員の指導を支援する。

参考文献

- 1) 村井万寿夫: 学習意欲を高めるための手立てについて, 日本教育工学会研究報告集, JET03-4, pp.31-36 (2003).
- 2) 倉田英和, 富永浩之, 林敏浩, 山崎敏範: 実行テストを用いたコンテスト形式の入門的 C プログラミング演習の大会運営サーバの開発, 情処研報, Vol.2006, No.108, pp.9-16 (2006).
- 3) 富永浩之, 倉田英和, 林敏浩, 安藤一秋, 垂水浩幸: コンテスト形式による初級 C プログラミングの演習支援, 情処研報, Vol.2008, No.42, pp.49-56 (2008).
- 4) 川崎慎一郎, 富永浩之: 競争型学習を取り入れた入門的 C プログラミング演習 - 演習支援サーバ tProgrEss の出題解答と採点結果のページ表示の改良 -, 信学技報, Vol.109, No.335, pp.187-192 (2009).
- 5) 倉田英和, 富永浩之, 林敏浩, 山崎敏範: グループコンテスト形式の C プログラミング演習支援環境 tProgrEss - 出題構造に基づいた入出力サンプルでの実行テスト -, 信学技報, Vol.106, No.507, pp. 87-92 (2007).
- 6) 川崎慎一郎, 富永浩之: 競争型学習を取り入れた入門的 C プログラミング演習 - 実行テスト系列による部分採点のための柔軟な照合機能 -, 情処研報, Vol.2009-CE-103, No.10, pp.1-8 (2010).
- 7) 富永浩之, 川崎慎一郎: 競争型学習を取り入れた入門的 C プログラミング演習 - 運用実験での実行テスト系列の利用状況 -, 情処研報, Vol.2010- CE-104, No.4, pp.1-12 (2010).
- 8) 川崎慎一郎, 富永浩之: 競争型学習を取り入れた入門的 C プログラミング演習 - 授業実践における解答コードの提出履歴の詳細分析 -, 情処研報, Vol.2010-CE-107, No.10, pp.1-10 (2010).
- 9) 西村智治, 川崎慎一郎, 富永浩之: 競争型学習を取り入れた初級 C プログラミング演習における教師支援 - モニタリング機能としての解答状況と提出状況の視覚化 -, 信学技報, Vol.110, No.453, pp.163-168 (2011).