

プログラミング学習カルテの分析による 人的教育資源の有効活用

田口 浩^{†1} 原田 史子^{†2}
高田 秀志^{†2} 島川 博光^{†2}

本論文では、プログラミング教育において、多数の学習者を受け持つ教員に、限られた時間での教育効果を高めるために有用な情報を提供する手法を提案し、それを大学教育に適用した結果を示す。本手法では、演習課題の出来具合に基づき、教員が個々の学習者の理解状況を容易に把握できるプログラミング学習カルテを実現する。プログラミング学習カルテにより、学習者の学習経過を反映した技法ごとの理解状況をとらえられる。さらに、プログラミング学習カルテを分析し、行き詰まりのおそれが高い学習者を優先的に、理解状況が似ている学習者をまとめて指導可能にする情報も生成して提供する。本手法に基づく学習カルテシステムを構築し、大学のC言語プログラミング教育に適用した結果、適切な指導支援情報を得ることができた。

Effective Utilization of Human Resources with Personal Learning Records in Programming Education

HIROSHI TAGUCHI,^{†1} FUMIKO HARADA,^{†2}
HIDEYUKI TAKADA^{†2} and HIROMITSU SHIMAKAWA^{†2}

In this paper, we propose a method to utilize human resources effectively in programming education. The method makes personal learning records from programming exercise results. The records represent learning context with the understanding level and the amounts of learning efforts for each item. Furthermore, the method analyzes the records and provides supporting information for teachers. It picks out students requiring supplemental supervision, using records of other students who took the same programming course in the past. In addition, it divides the students into groups depending on their learning contexts. We have implemented a system based on the method and applied it to an actual C programming course in a university. The results have proved that the method can generate effective supporting information.

1. はじめに

情報通信技術は今や、身近な家電製品や携帯電話から産業、行政、社会の基幹システムに至るまで、あらゆる場面で活用されている。このような高度情報通信社会の著しい発展にともない、それらを支える人材の需要がきわめて大きくなっている。しかし、現在、高度な情報通信技術者は質、量ともに深刻な不足状態にあるといえる。そのため、大学をはじめとする高等教育機関には、情報系の専門を選択した学生に対して高い水準の情報通信教育を行い、社会に大量かつ急速に良質な人材を供給することが求められている。

産業界は、情報通信技術者を目指す学生は在学中にプログラミングを習得しておくべきとしている¹⁾。プログラミングは情報通信のどの分野においても基礎的な技術となる。それゆえ、大学の情報系学科では低回生時からプログラミング教育を実施し、プログラミング能力の育成に努めている。しかし、カリキュラムの終了時点で期待されるレベルに到達できていない学生が相当数にのぼっていることは否定できない。

大学におけるプログラミング教育がうまくいかない原因の1つとして、学習者の数や教えるべき内容と量に比べて、教える側の人的資源が非常に限られていることがあげられる。そのため、行き詰まっている学習者に対して教員のフォローアップが十分に行えず、脱落する学習者が増える。教員の数を増やせば、脱落する学習者を減らすことができるが、現実的な解とはいえない。このような状況では、学習者全体のプログラミング能力を最大限に向上させるという視点で、限られた人的教育資源を有効活用することが最善策と考えられる。

本論文では、プログラミング教育において、教員が多くの学習者に対して効率的に指導を行うための指導支援情報を提供する手法を提案する。我々の手法では、教員が個々の学習者の詳細な理解状況を容易に把握できるプログラミング学習カルテを実現する。さらに、プログラミング学習カルテを分析して指導に有益な情報を導き出し、教員に提供する。本手法は、

- プログラミング演習課題の出来具合から学習者の理解状況を定量的に求める、
- 過去の学習者のデータに基づいて行き詰まりのおそれが高い学習者を抽出する、
- 抽出された学習者を理解状況が似ている学習者ごとにグルーピングする、

^{†1} 三菱電機株式会社先端技術総合研究所
Advanced Technology R&D Center, Mitsubishi Electric Corporation

^{†2} 立命館大学情報理工学部
Department of Information Science and Engineering, Ritsumeikan University

という特長を持ち、得られる情報によって、教員が学習者の指導に使う時間を有効活用させる。教員は、行き詰まりのおそれが高い学習者を優先的に指導でき、似た問題をかかえる学習者をまとめて指導することが可能になる。さらに、指導する学習者がいかなる点でつまづいているかを容易に把握できる。したがって、教員は適切な指導を行うまでに必要な作業や、複数の学習者への同じ指導の繰返しといったことにかかる時間と負荷を削減でき、つまづきを解消するための指導時間を増やすことができる。我々は、本手法に基づくプログラミング学習カルテシステムを構築した。本システムを用いてプログラミング演習を行うことにより、教員は少ない負荷で指導支援情報を得ることができる。本手法を大学の C 言語プログラミング科目に適用した結果、早い時点で行き詰まりのおそれが高い学習者の抽出とグルーピングを適切に行うことができた。

以下、2章でプログラミング技法の教育と学習のそれぞれにおける特徴について述べる。3章ではプログラミング学習カルテの実現と分析の手法を説明し、4章で本手法に基づくプログラミング学習カルテシステムについて述べる。5章では大学の C 言語プログラミング科目に本手法を適用した結果を示す。そして、6章で関連研究との比較を行ったうえで、7章でまとめを行う。

2. プログラミング技法の教育と学習

2.1 大学におけるプログラミング教育の特徴

大学におけるプログラミング教育では、個々の技法ごとに、講義形式で教員が概念や原理、実現法を説明した後に、学習者が実際にその技法を用いたプログラミングの演習を行う。講義では教員が全学習者に向けていっせいに説明を行うのに対し、演習では学習者が各自で演習課題に取り組み、教員が各学習者の状況に応じて適宜個別に指導する。これにより、習得が遅れている学習者をフォローアップし、学習者全体のプログラミング能力の向上を図る。しかし、この個別のフォローアップが十分に行えていない教育現場は少なくない。

演習は講義に比べて授業クラスの規模が小さく設定されるが、それでも1名の教員が数十名のクラスを担当することは珍しくない。演習には教員を補助する TA (Teaching Assistant) が配置されることが多いが、TAには教員に準ずる高い指導力が求められ、それを満たす人材の数も限られている。また、大学や教員によってはオフィスアワーが設けられる場合があるが、教員が個々の学習者と対面して指導できるのは授業時間内が主となる。このように教員が学習者1人ひとりの指導にかけられる時間が非常に限られていることが、上述の問題の原因になっているといえる。

教える側の人的資源が限られているにもかかわらず個別指導にこだわると、指導時間を確保できた一部の学習者に対しては高い指導効果を与えられるが、時間を確保できなかった学習者に対してはまったくフォローアップを行えない。このように一部の学習者に指導効果が偏るよりも、指導を必要とする学習者にはできるだけ公平に指導を与えるほうが、学習者全体のプログラミング能力を向上させるために有効であると考えられる。ここで、指導を必要とする学習者とは、自ら積極的に指導を求める学習者だけではなく、指導を行わなければ脱落のおそれがある学習者も含む。教員が指導を与えられる学習者の数を増やすためには、学習者1人あたりにかかる指導時間を減らさなければならない。その解決策として、完全な個別指導ではなく、理解状況が似ている学習者同士をまとめた小グループでの指導とすることが現実的かつ効果的であると考えられる。

人的教育資源が限られた教育現場の支援法を考えるうえで、着目すべき点がある。プログラミング教育では、対象とするプログラミング言語が同じであれば、教える内容のおおかたは一般的に決まっている。したがって、年度が異なる学習者間で学習する内容に大きな差異は生じない。このことをふまえると、個々の学習者の学習経過を記録して分析できる仕組みがあれば、教員が次の学習者を指導する際に有用な情報を提供できると考えられる。

2.2 C 言語プログラミングの学習内容

代表的なプログラミング言語である C 言語は、大学の情報系学科において、最初のプログラミング言語として教えられることが多い。C 言語を初めて学ぶ学習者はまず、C 言語特有の概念や文法といった基礎技法を学習する。基礎技法をひととおり学習した学習者は次に、一般的な問題を解決するデータ構造やアルゴリズムを C 言語で実現する技法を学ぶ。それらの技法を基礎技法と区別して、応用技法と呼ぶこととする。応用技法では、データ構造あるいはアルゴリズムの概念や原理を理解したうえで、基礎技法をいくつも組み合わせて実現することが求められる。それら多数の要素の1つ1つを深く理解していなければ、プログラムとして実現することはできない。そのため、基礎技法の学習に比べて習得の難しさが格段に増し、学習の途中で行き詰まる学習者が後を絶たない。

図1に示すように、基礎技法および応用技法としての一般的な学習内容は、文献2)や文献3)などに基づくところ、それぞれいくつかの単元に大きく分けられる。各単元の内容はさらに細かく分割することができ、細分化した学習内容を学習項目と呼ぶ。

2.3 プログラミング技法の理解過程

あるプログラミング技法を学んだとき、すぐにその技法をプログラムとして実現できる学習者もいれば、なかなか実現できない学習者もいる。このことから、学習者は学習量に比例

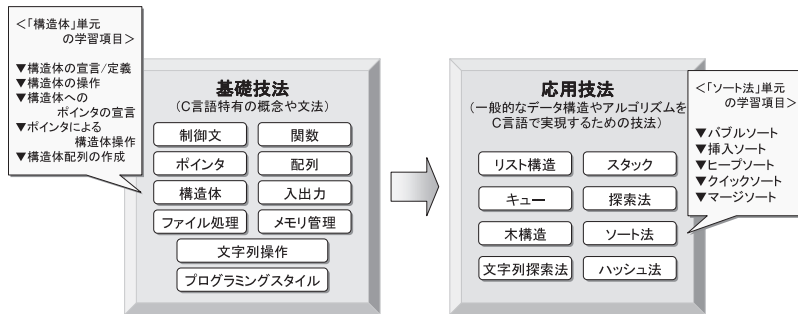


図 1 C 言語プログラミングの学習内容
Fig. 1 Contents of C programming exercise courses.

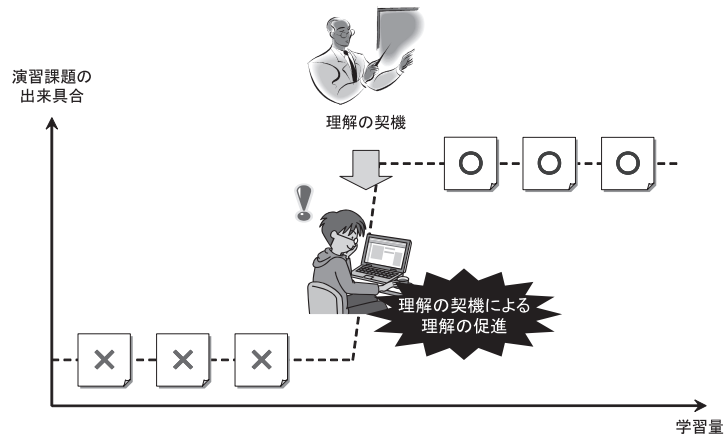


図 2 プログラミング技法の理解過程
Fig. 2 Understanding process of programming techniques.

して徐々に理解するのではなく、特定の事象により理解が大きく促進されると考えられる。我々はそのような事象を理解の契機と呼んでいる。

図 2 に示すように、学習者が自身にとっての理解の契機を教員や同僚、あるいは参考書などから与えられるまでは、その技法をうまく扱えない。一方、理解の契機が与えられて理解が促進された後は、その技法を容易に扱ってプログラムを作成できるようになる。したがって、演習において、学習者が多くの演習課題の中で正しく実現できている技法は深く理

解できているといえる。

各々の技法の学習において、唯一の事象がすべての学習者の理解の契機になるとは限らないが、多くの学習者に共通する理解の契機は存在する。我々は、教員による説明内容の中から、先に理解できた多くの学習者に共通する理解の契機を抽出して、同内容をまだ理解できていない学習者に提示するための手法を提案した⁴⁾。当手法を大学での開講科目に適用した結果、先に理解できた学習者に共通する理解の契機を与えることによって、理解できていなかった学習者の理解を促進させることができた。これより、理解状況が似ている学習者同士であれば、まとめて共通の指導内容を与えることで、効率的かつ効果的な指導が可能であると考えられる。

3. プログラミング学習カルテを用いた指導負荷の軽減

3.1 教員の 3 種類の負荷

大学でのプログラミング教育、特に演習科目における教員の負荷は、他の科目に比べて一般的に大きいといわれる。プログラミング演習における教員の主な作業は、

- 演習課題、および、その出来具合を評価するための基準の作成、
- 演習課題の出来具合の評価、
- 演習課題の出来具合に応じた個々の学習者に対する指導、

の 3 つと考えられる。各々の作業を行ううえで教員にかかる負荷をそれぞれ、課題作成負荷、評価負荷、指導負荷と呼ぶこととする。教員のこれらの負荷を軽減させることができれば、現状よりも充実したプログラミング教育を実現できる。

図 3 に示すように、演習課題および評価基準の作成は科目の準備段階に行う作業であるので、課題作成負荷は開講前にのみ生じ、負荷の大きさは学習者の数によらない。一方、評価ならびに指導は開講中、つねに繰り返す行う作業であり、受け持つ学習者が多くなるとつれて作業量も相当大きくなる。しかし、評価作業については高い指導力を必要としないので、客観的かつ明確な演習課題の評価基準を作成すれば、プログラミングを習得している上回生を評価作業として多数配置するなどの対処により、教員の評価負荷を分散させることが可能である。これに対し、指導作業については担当する者の指導力が指導の質に大きく影響するので、教員の指導負荷を人的資源の投入によって分散させることは容易ではない。

本論文では、教員の指導負荷を軽減させることを目的として、多くの学習者を受け持つ教員が効率的に指導を行えるような指導支援情報を提供する手法を提案する。提案する手法は、学習者の学習の経過をつぶさに記録し、教員が 1 人ひとりの理解状況をつねに把握でき

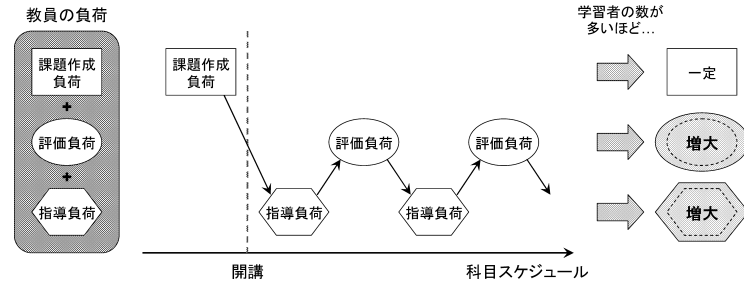


図3 プログラミング演習における教師の負荷

Fig. 3 Teacher workloads in programming exercise courses.

るように管理しておく。そのうえで、それらの記録を分析して、行き詰まるおそれの高い、優先して指導すべき学習者を抽出する。さらに、抽出した学習者の中で、理解状況が似ている者同士をグルーピングし、それぞれのグループの特徴を教員に提示する。これにより、適切な指導を行うまでに必要な作業や、複数の学習者への同じ指導の繰返しといったことにかかる時間と負荷を削減させ、限られた人的教育資源を有効活用した指導を実現する。

3.2 プログラミング学習カルテの実現

医療の分野では、医師は患者の診療に関する経過を、一般的にカルテと呼ばれる診療録に必ず記録する。ある症状や疾患に対する治療法はただ1つに決まらない場合がほとんどであり、医師にはその時々に応じて適切な判断が求められる。カルテに時系列に記録を残すことにより、医師は患者の過去の診療情報も参考にしながら、現在の症状に対するより適した治療方針を立てることができる。また、カルテによって医師間で診療情報の共有が行われる。

2.2節で述べたとおり、C言語の応用技法のようなプログラミング技法の習得には、関連する種々の学習項目の理解が必要となる。それゆえ、学習者がそのような技法の習得に行き詰まっている場合、その原因は多数考えられる。たとえば、リスト構造を実現するプログラムの作成で行き詰まっているならば、リスト構造の概念や原理、ポインタに関する学習項目、構造体に関する学習項目、メモリ管理に関する学習項目のうちのいずれかが1つ、またはいくつかの理解不足が原因として考えられる。教員は原因が分かれば、それに応じた指導を行って行き詰まりを解消させられる。学習者自身が原因を認識していることはほとんどないので、教員が学習者の理解状況から真の原因を突き止める必要がある。しかし、教員が多数の学習者の指導を同時に担当しているとき、それぞれの学習者の理解状況をつねに的確に把握し続けることは困難である。そのため、学習者が行き詰まった際に、行き詰まりを解消す

るための指導よりも理解状況の把握や原因の追究のために、教員が多くの時間を費やすことがよくある。以上のことをふまえると、医療カルテのように、プログラミング学習者の学習経過を細かく記録しておくことで、教員による指導を改善、充実させることが期待できる。そこでまず、教員が個々の学習者の理解状況を必要時に把握できるプログラミング学習カルテの実現を考える。

我々は、学習者がこれまでにどれだけ学習に取り組み、そして、現時点でどの程度理解できているか、の2点を学習項目ごとに明らかにすることで理解状況を表現する。本論文では、前者を取り組み度合、後者を理解度合と呼ぶ。この独立した2つの指標を示すことにより、単に理解できているか、できていないかだけでなく、多くの時間をかけて学習しているのに理解できていないのか、それとも、積極的に学習に取り組んでいないから理解できていないのか、といったことも分かる。学習者は、講義で説明された技法を、演習においてプログラミング演習課題に取り組むことで体得する。このことから、演習課題の出来具合は学習者の理解状況を反映しているといえる。それゆえ、提案手法では、取り組み度合と理解度合を求めるために、演習課題の出来具合を示すデータを取得する。

演習課題の出来具合を判定するためには、プログラムが正しく動作するかどうかだけでなく、個々の技法が正しく実現されているかどうかまで細かく評価しなければならない。本手法では、演習課題ごとに、実現する内容を細かく区切った評価観点をあらかじめ10~20個ずつ設ける。そして、評価観点ごとに、その内容を実現するために用いる学習項目を対応付ける。さらに、各評価観点の当該演習課題における重要度を1~10の10段階程度で定めておく。演習課題と評価観点を例を表1に示す。教員はそれぞれの評価観点について、「実現できている」、「実現できていない」、「どちらともいえない」のいずれかを判定することによって学習者の出来具合を評価する。「どちらともいえない」と判定されるのは、評価観点到示されている内容とは異なる方法で実現している場合や、できているか判断し兼ねる場合である。これにより、詳細かつ客観的に演習課題の出来具合を評価することができる。

図4に示すように、以上の方法による演習課題の評価結果を随時取得し、蓄積しておく。そして、教員が学習者の理解状況を知りたいときに、それらのデータから学習項目ごとの取り組み度合と理解度合を求め、動的にプログラミング学習カルテを構成する。

3.3 取り組み度合と理解度合の定量表現

多数の学習者を受け持つ教員にとって有用なプログラミング学習カルテを実現するためには、すべての学習者の理解状況を統一した基準で表現する必要がある。このことをふまえ、学習者が取り組んだ演習課題の評価結果に基づいて、学習項目ごとの取り組み度合と理解度

表 1 評価観点の例
Table 1 Example of evaluation criteria.

番号	評価観点	学習項目	重要度
1	ファイルを読み込み用にオープンする	ファイルのオープン	6
2	ファイルのオープン時にエラー処理を行う	ファイルのオープン	6
3	ファイルの終端までの繰り返し処理を行う	ループ (for 文/while 文)	6
4	ファイルからデータを読み込む	文字列の入力, ファイルからの入力	6
5	ファイルをクローズする	ファイルのクローズ	6
6	ファイルの終端を検出する	ファイルの終端判定	6
7	標準出力ヘデータを書き出す	文字列の出力, 標準出力への出力	2
8	双方向リスト用の構造体を定義する	双方向リストの実現	10
9	双方向リストへ新たな要素を追加する	双方向リストへの要素の追加	10
10	malloc() を用いて各要素が必要とするメモリを確保する	メモリ領域の確保	8
11	プログラムの最後で各要素が確保していたメモリ領域を解放する	終了時のメモリ解放	8
12	プログラム中に適切なコメントを付ける	コメントの記述	4

合を定量的に求める。

プログラミング技法の習得に向けた学習の取り組み量は、各々の学習項目を用いたプログラミングをどれだけ行ったかに等価であると考えられる。よって、学習項目の取り組み度合は、その学習項目が対応付けられた評価観点を含む演習課題を解いた数を基準にして算出する。このとき、それぞれの演習課題において、評価観点ごとに重要度および難易度が異なることに着目する。まず、重要度と学習者の取り組みの関係を考える。各々の演習課題には、それを出題する主たる目的となる学習項目が存在する。学習者がその演習課題によって新しく学ぶような学習項目が該当し、それらに対応する評価観点の重要度は高くなる。演習課題を完成させるためには、主要な学習項目以外の学習項目も用いる必要がある。そのような

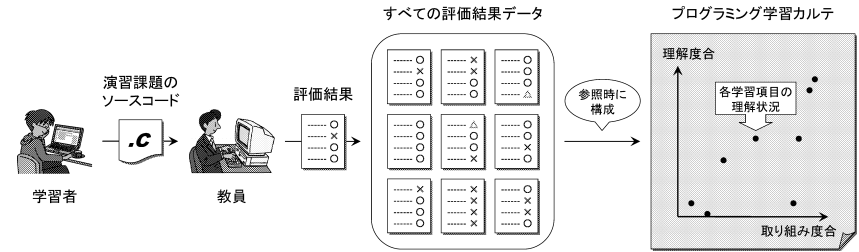


図 4 プログラミング学習カルテの実現
Fig. 4 Realization of personal learning records.

学習項目は、学習者が既習のものである場合がほとんどであり、それらに対応する評価観点の重要度は低くなる。ゆえに、重要度が高い評価観点ほど、学習者が対応付けられた学習項目を理解するうえで、その内容に取り組むことの意義が大きいと見える。一方、難易度と学習者の取り組みの関係は、難易度が高い評価観点ほど、その内容の実現方法について学習者が深く考えた可能性が高いと考えられる。ゆえに、難易度が高い評価観点の内容に取り組んだ場合のほうが、難易度が低い場合に比べて、取り組みの意義が大きいと見える。重要度は演習課題の作成者によって設定されるのに対し、難易度は学習者全体の正答率によって客観的に決めることができる。

以上をまとめると、評価観点ごとに取り組む意義は異なり、重要度と難易度のそれぞれが高いほどその意義は大きくなる。よって、各評価観点の取り組み意義の大きさを明確にすることで、学習の取り組み度合を的確に求められる。本手法では、取り組み意義の大きさを数値化したものをエフォートスコアと定義し、評価観点ごとにエフォートスコアを設定する。エフォートスコアを1~5の5段階で設定する場合の設定例を表2に示す。学習者のある時点での取り組み度合は、学習項目ごとに、学習者がそれまでに解いた演習課題の評価観点の中で、当該学習項目が対応付けられている評価観点のエフォートスコアを、評価結果にかかわらずすべて足し合わせて求める。

次に、理解度合の求め方を考える。2.3節で述べたように、多くの演習課題で実現できている学習項目ほど、学習者がよく理解できている可能性が高いといえる。また、取り組みの意義とは異なり、実現できていることの意義は重要度や難易度によらず、いずれの評価観点においても同じであると考えられる。よって、本手法では、学習項目ごとに、それが対応付けられた評価観点の評価結果が「実現できている」である演習課題の数をそのまま用いて、学習者の理解度合を算出する。これにより、取り組み度合とは独立した指標として、理解度

表 2 エフォートスコアの設定例
Table 2 Example of the effort score.

		難易度		
		高	中	低
重要度	高	5	4	3
	中	4	3	2
	低	3	2	1

合を求められる。

このようにして求められた取り組み度合と理解度合を用いることで、学習項目単位での学習経過を反映した理解状況を表現できる。それゆえ、教員が容易に学習者の理解状況を把握できるプログラミング学習カルテが実現される。

3.4 優先して指導すべき学習者の抽出

学習者は、何度も理解に行き詰まったり、行き詰まりがなかなか解消されなかったりすると、学習意欲が低下し、挫折するおそれが高まる。それゆえ、教員はそのような状態に陥っている、あるいは、陥りそうな学習者を早期に見つけ出して優先的に指導することが求められる。

プログラミング学習カルテの実現により、学習者間で理解状況を定量的かつ容易に比較することが可能になる。同時に学習を進めている学習者同士の比較だけでなく、現在学習中の学習者と同じ方法で学習経過を記録した過去の学習者との比較も行える。過去の学習者については、当該カリキュラム終了時の理解状況が明らかになっている。そのうえ、学習中の任意の時点での理解状況も求められる。そこで、現在の学習者と過去の学習者の理解状況を比較することで、優先的に指導すべき学習者を抽出する。なお、理解状況は演習課題の取り組み結果に基づいて求められるので、授業スケジュールが大きく変わらず、かつ、同一基準で演習課題の評価観点の作成ならびに学習項目の対応付けが行われれば、講義や演習の内容の一部が変更または改善されても、過去の学習者と理解状況を比較することができる。

本手法では、学習者間の理解状況を比較し、両者の類似度を定量的に算出する。類似度を求めるうえで、学習者のある時点での理解状況を空間上の点で表現する。ここで考える空間は、各学習項目の取り組み度合と理解度合のそれぞれの値の軸で構成される（学習項目数 × 2）次元の空間である。ある時点におけるすべての学習項目の取り組み度合と理解度合の値の組を座標とする点は、この空間上の1つの点として表現される。比較する両学習者の理解状況を表す点をこの空間上にそれぞれ配置し、2点間のユークリッド距離を求める。そし

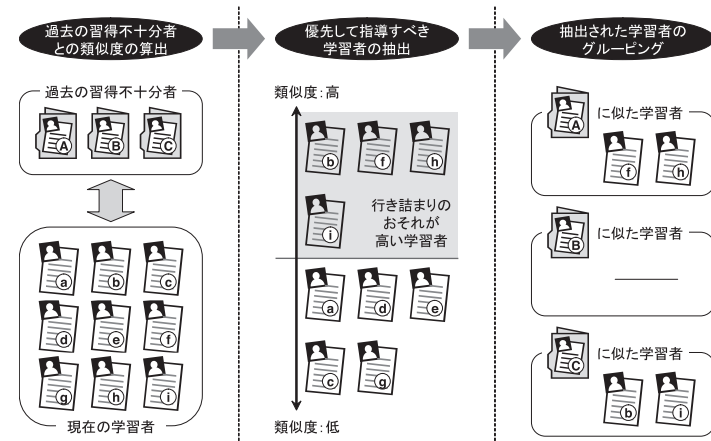


図 5 プログラミング学習カルテの分析
Fig. 5 Analysis of personal learning records.

て、その距離が小さいほど、両者の類似度が高いと見なす。

以上の方法により、ある時点での理解の状態だけでなく、それまでの学習経過も考慮した比較が行える。これを用いて、図 5 に示すように、過去の学習者の中で結果的に習得が不十分であった学習者と現在の各学習者の理解状況の類似度を調べ、類似度が高い学習者を抽出する。本論文では、対象のカリキュラムの終了時点で習得が不十分である学習者を習得不十分者と呼ぶ。具体的には、学習者ごとに、過去の習得不十分者 1 人ひとりとの類似度を求め、そのうち最も高いものを当該学習者の類似度とする。このとき、過去の学習者については、現在の学習者と同じ時点の理解状況を求めて用いる。過去の習得不十分者は、いずれかの時点で理解に行き詰まったと考えられる。理解に行き詰まりはじめた学習者は、順調な学習者に比べて学習が進まなくなるので、取り組み度合および理解度合にその差が表れる。そのような過去の学習者と理解状況が類似していることは、すでに当該学習者に順調な学習者との差が表れていることを示しており、教員の指導を必要としている可能性が高いといえる。したがって、そのような学習者をなるべく早い段階で抽出して教員に提示することにより、教員は行き詰まりのおそれが高い学習者を優先して指導できる。

3.5 理解状況に基づく学習者のグルーピング

優先して指導すべき学習者が同時に複数存在する場合、教員が指導すべき内容がその全員

に対して同じであるとは限らない。それゆえ、それらの学習者を理解状況が似ている学習者ごとにグルーピングすることで、教員が多くの学習者に対してより効率的に指導を行える。過去の習得不十分者についても、その原因は学習者ごとに異なる。そこで、現在の学習者の中から優先して指導すべき学習者を抽出する際に、過去のどの学習者との類似度が最も高いかを求め、それに応じて優先して指導すべき学習者をグルーピングする。これにより、理解状況が似ている学習者同士をグルーピングできる。さらに、各グループの特徴は、学習項目ごとの理解状況を分析することで明らかになる。他のグループに属する学習者に比べて、取り組み度合および理解度合が低い学習項目があれば、その学習項目の理解不足あるいは取り組み不足が行き詰まりの原因になりうると推測される。したがって、そのような各グループの理解状況の特徴を、グルーピングの結果とあわせて教員に提示することで、教員は理解状況が似ている学習者同士をまとめて指導できるようになり、さらにその指導指針を立てやすくなる。グループ単位ではなく個人指導を必要とする際にも、教員は個々のプログラミング学習カルテを参照することで、より個人に合わせた指導を行える。

このように本手法は、プログラミング教育で必ず実施される演習課題の評価結果からプログラミング学習カルテを構成し、それを分析して教員の指導を支援するための情報を提供する。提供される情報の活用により、理解状況の把握や優先して指導すべき学習者の同定、複数の学習者への同じ指導の繰返しといったことにかかる時間と負荷を削減でき、その分だけ指導の質を改善、充実させることが可能になる。また、学習者全体のプログラミング学習カルテを俯瞰することで、全体的に理解度合または取り組み度合が低い学習項目を明らかにできるので、講義や演習の内容の改善にも役立てられる。ゆえに、本手法はプログラミング教育における限られた人的教育資源の有効活用を実現する。

4. プログラミング学習カルテシステム

4.1 システムの概要

提案する手法に基づき、学習者の学習経過を記録してプログラミング学習カルテを実現するシステムを構築した。図6にその構成を示す。本システムは、Java サーブレットで実装された Web アプリケーションであり、学習者と教員に対してそれぞれ異なるインタフェースを用意している。両インタフェースは任意のクライアント PC 上の Web ブラウザから利用でき、授業時間外でも利用可能である。

本システムの利用に先立ち、教員が演習課題ならびにその評価観点を登録する。その際に、各々の評価観点について、重要度および対応する学習項目もあわせて登録できる。

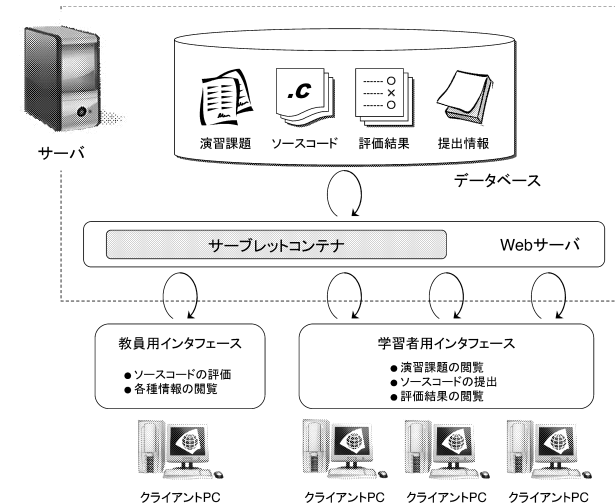


図6 プログラミング学習カルテシステムの構成

Fig. 6 Configuration of the system based on the method.

学習者用インタフェースでは、演習課題の内容確認と、作成したソースコードとその実行結果の電子提出ができる。提出時には、演習内容に対する理解度の自己評価と、自由記述のコメントを登録可能である。これにより、学習者自身が認識している理解状況を教員に伝えることができる。また、教員による評価結果も閲覧でき、効果的な復習が行える。

教員用インタフェースでは、提出されたソースコードの評価ができる。評価の際には、図7に示すように、学習者が提出したソースコードと実行結果に加え、学習者の自己評価およびコメントが表示される。システム上でそれらを参照して、3.2節で述べた方法で評価を行い、その結果を直接登録する。教員も評価登録時にコメントを登録できるので、ここで学習者のコメントに対する返答も可能であり、本システムが学習者と教員とのコミュニケーションを補助する役割も果たすことが期待される。

4.2 教員への指導支援情報の提供

本システムでは、教員が登録した評価観点ごとの評価結果のデータをそのままの形で管理する。教員が学習者の理解状況を参照したいときに、それらのデータから学習項目ごとの取り組み度合と理解度合を求め、最新のプログラミング学習カルテを構成する。教員は教員用インタフェースからその学習カルテを参照できるので、対面指導を行う際に手元にコンピュー

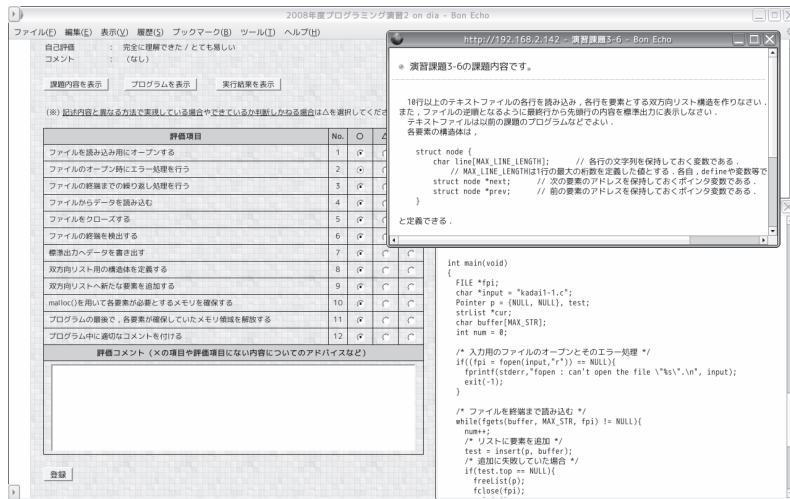


図 7 ソースコード評価時のインターフェース例
Fig. 7 Interface for source code evaluation.

タがあれば、その場で学習カルテを参照することが可能である。したがって、教員は演習課題の評価作業を本システム上で行うだけで、学習者の理解状況をつねに把握できるようになるとともに、提案手法を用いた分析が実行可能になる。

本手法によって提供されるのは、教員が学習者全体に対して効率的に指導を行うための支援情報である。それらの情報とは別に、本システムでは、個々の学習者について、提出状況ならびに演習課題ごとのソースコードやコメント、評価結果といった詳細な情報を参照できる。それゆえ、本システムの導入により、教員は少ない負荷で効率的かつ最適な指導を行うことができる。

5. 評価

実際に大学で開講されている C 言語プログラミング科目において取得したデータを用いて、本手法の有効性を検証した。

5.1 実現場でのデータ取得

検証の対象とした科目は、立命館大学情報理工学部情報システム学科で 2007 年度後期に開講された 1 回生配当の演習科目「プログラミング演習 2」である。この科目では、前期開

表 3 2007 年度「プログラミング演習 2」の授業スケジュール
Table 3 Course schedule of “Programming Exercise 2” in 2007.

週	テーマ	演習内容
1	データ構造とアルゴリズムの基礎 (1)	基礎技法の復習
2	データ構造とアルゴリズムの基礎 (2)	基礎技法の復習, メモリ管理
3	リスト構造 (1)	単方向リスト
4	リスト構造 (2)	双方向リスト
5	スタックとキュー	スタック, キュー
6	探索法 (1)	リニアサーチ
7	探索法 (2)	バイナリサーチ
8	木構造 (1)	木構造, 二分探索木
9	これまでの復習	—
10	木構造 (2)	深さ優先探索, 幅優先探索
11	中間試験	—
12	ソート法 (1)	挿入ソート, バブルソート, ヒープソート
13	ソート法 (2)	クイックソート, マージソート
14	文字列探索法	KMP 法, BM 法
15	ハッシュ法	ハッシュ法

講科目において、C 言語プログラミングの基礎技法のうち、メモリ管理単元を除く各単元を一ひとり学んだ学生を対象に、メモリ管理単元および応用技法の各単元についてのプログラミング演習を行う。なお、同学科では、当科目で演習するプログラミング技法を説明する講義科目「データ構造とアルゴリズム」が同時期に当科目と連動して開講された。表 3 に、当科目の授業スケジュールを示す。受講者には各週 1~2 題の必須課題に加え、1~7 題の選択課題を出題し、すべての必須課題と 1 題以上の選択課題に取り組み、作成したソースコードを提出するよう求めた。146 名の学生が 4 クラスに分かれて受講し、各クラスには教員 1 名と大学院生の TA が 5~6 名が配置され、教員と TA が分担してソースコードの評価および受講者への指導を行った。当科目において、すべての受講者、教員および TA にプログラミング学習カルテシステムを用いてソースコードの提出および評価の作業を行ってもらい、評価結果のデータを取得した。ここで取得したデータは、提案手法による指導支援情報を提供していない場合のデータである。

本手法を適用するためには過去の学習者のデータが必要となる。そこで、2006 年度後期開講の同科目においても、同様の方法でデータを取得した。2006 年度と 2007 年度とでは復習にあてた週が異なるが、両年度ともほぼ同じスケジュールで開講され、出題した演習課題もすべて同じである。

表 4 指導優先度の判定

Table 4 Judgement of students requiring supplemental supervision.

判定時点	習得不十分者 29 名の分布					A のみ抽出		A と B を抽出	
	A	B	C	D	E	再現率	適合率	再現率	適合率
第 2 週開始時	23 名	0 名	2 名	4 名	0 名	—	—	79.3%	39.0%
第 3 週開始時	16 名	9 名	2 名	2 名	0 名	55.2%	53.3%	86.2%	42.4%
第 4 週開始時	17 名	8 名	1 名	3 名	0 名	58.6%	56.7%	86.2%	42.4%
第 5 週開始時	17 名	7 名	2 名	3 名	0 名	58.6%	56.7%	82.8%	40.7%
第 6 週開始時	17 名	6 名	5 名	1 名	0 名	58.6%	56.7%	79.3%	39.0%
第 7 週開始時	17 名	8 名	3 名	1 名	0 名	58.6%	56.7%	86.2%	42.4%
第 8 週開始時	17 名	8 名	4 名	0 名	0 名	58.6%	56.7%	86.2%	42.4%
第 9 週開始時	17 名	8 名	4 名	0 名	0 名	58.6%	56.7%	86.2%	42.4%
第 10 週開始時	22 名	3 名	4 名	0 名	0 名	75.9%	73.3%	86.2%	42.4%
第 11 週開始時	21 名	5 名	3 名	0 名	0 名	72.4%	70.0%	89.7%	44.1%
第 12 週開始時	21 名	6 名	2 名	0 名	0 名	72.4%	70.0%	93.1%	45.8%
第 13 週開始時	20 名	5 名	4 名	0 名	0 名	69.0%	66.7%	86.2%	42.4%
第 14 週開始時	20 名	8 名	1 名	0 名	0 名	69.0%	66.7%	96.6%	47.5%
第 15 週開始時	25 名	3 名	1 名	0 名	0 名	86.2%	83.3%	96.6%	47.5%

5.2 指導優先度の判定

はじめに、取得したデータを用いて、優先して指導すべき受講者を早い段階で抽出できるかどうかを検証した。今回の検証では、第 10 週以前の学習内容で挫折した、すなわち、第 11 週以降の演習課題にまったく取り組んでいなかった受講者を習得不十分者と見なした。この基準に該当する受講者は、2006 年度は 41 名、2007 年度は 29 名いた。2007 年度の全受講者の中から、この 29 名を優先して指導すべき受講者として早期に同定できたかを調べた。

表 4 は、各週の授業開始時点において、各々の受講者の優先指導の必要性を示す優先度を A～E の 5 段階で判定したときに、習得不十分者 29 名の分布を示している。優先度 A が最も優先指導の必要性が高く、優先度 E が最も低い。また、優先して指導すべき受講者として抽出する基準を、優先度 A と判定された受講者のみとする場合と、優先度 A または B と判定された受講者とする場合について、習得不十分者の抽出における再現率と適合率もそれぞれ示している。再現率は習得不十分者 29 名のうち抽出された者の割合を、適合率は抽出された受講者の中の習得不十分者の割合を表す。優先度の判定のために、2007 年度の全 146 名の受講者について、各時点での、2006 年度の習得不十分者 41 名それぞれとの類似度を、3.3 節で述べた方法で空間上のユークリッド距離として求めた。そして、求めた 41 個の距離のうちの最短距離が小さい順に全受講者を 5 等分し、順に優先度 A～E とした。このとき、優先度 A のみを 30 名とし、優先度 B～E はそれぞれ 29 名ずつとした。

表 5 エフォートスコアの設定基準

Table 5 How to set the effort score.

	重要度	高 (8~10) 中 (4~7) 低 (1~3)	難易度 (r は正答率)		
			高 ($r \geq 96.0\%$)	中 ($96.0\% > r \geq 88.0\%$)	低 ($88.0\% > r$)
必須課題	高	(8~10)	5	4	3
	中	(4~7)	4	3	2
	低	(1~3)	3	2	1
選択課題	高	(8~10)	10	8	6
	中	(4~7)	8	6	4
	低	(1~3)	6	4	2

ここで、類似度を求めるうえで必要な各値の設定について説明する。表 5 は、取り組み度合を算出するために評価観点ごとに設定したエフォートスコアの値を示している。重要度は、課題作成者によって 1~3 が設定されている場合は「低」、4~7 が設定されている場合は「中」、8~10 が設定されている場合は「高」とした。難易度は、受講者の正答率が 88.0%未満の場合は「低」、88.0%以上 96.0%未満の場合は「中」、96.0%以上の場合は「高」とした。難易度の区分は、2006 年度の正答率に基づき、全評価観点をおよそ 3 等分にするような範囲を設定した。全員が必ず取り組む必須課題の評価観点には、重要度と難易度に応じて 1~5 の 5 段階でエフォートスコアを設定した。当科目で設けられている選択課題は、受講者が力量に応じて自主的に取り組むもので、必須課題よりも取り組みの意義が大きい。それゆえ、選択課題の評価観点には、必須課題の基準の 2 倍のエフォートスコアを設定した。一方、理解度合は、当該学習項目が実現できている必須課題の数に 5 を乗じた値と、選択課題の数に 10 を乗じた値を足し合わせて算出した。類似度を求める際に、取り組み度合と理解度合の重みが大きく変わらないようにするため、実現できている演習課題の数にこれらの値を乗じることとした。また、必須課題については、「データ構造とアルゴリズム」において解法の説明が行われたため、必須課題と選択課題とは実現できていることの意義の大きさが異なると思われ、それぞれに異なる値に差を持たせた。なお、我々は今回、基礎技法として 47 個、応用技法として 36 個の学習項目を定義した。

表 4 から分かるように、週が進むにつれて、より多くの習得不十分者を優先指導の必要性が高いと判定している。優先して指導すべき受講者として優先度 A と判定された受講者のみを抽出する場合は、第 8 週までの前半の各時点における再現率は 55.2~58.6%、適合率は 53.3~56.7%であった。第 2 週開始時点では、演習課題をまだ 1 題も提出していない受講者が全体の 39.7%を占め、それらの学習者は類似度が等しく、すべて優先度 A と判定さ

表 6 第 5 週開始時点におけるグルーピング結果
Table 6 Grouping results at the beginning of 5th week.

グループ	人数	取り組み度合が 0 もしくは 0 に近い	理解度合が 0 もしくは 0 に近い		理解度合が低い	
			取り組み度合が低い	取り組み度合が低くない	取り組み度合が低い	取り組み度合が低くない
5A	2 名	メモリ, 応用全般	—	—	—	コメント
5B	7 名	メモリ, 応用全般	改行	—	—	—
5C	3 名	メモリ, 応用全般	—	標準出力	—	列出力, コメント
5D	8 名	構造体, fread(), メモリ, 応用全般	改行	—	コメント	f 出力
5E	8 名	構造体, fread(), メモリ, 応用全般	改行	—	—	—
5F	2 名	コマンド行, 構造体, 標準出力, fread(), 応用全般	—	メモリ	ループ, 列出力, f 出力, f 終端	f 開
5G	1 名	コマンド行, 標準出力, 応用全般	—	—	列入出力, fp, f 入力	—
5H	3 名	コマンド行, 構造体, 標準出力, fread(), メモリ, 応用全般	コメント	—	列入出力, fp, f 開閉, f 入力	—
5I	4 名	コマンド行, 構造体, 標準出力, fread(), メモリ, 応用全般	—	—	列入出力, fp, f 開閉, f 入力	—
5J	1 名	構造体, fread(), メモリ, 応用全般	改行	コマンド行, 標準出力	—	列入出力, fp, f 開閉, f 入力
5K	3 名	ループ, コマンド行, 構造体, 標準出力, 列入出力, f 終端, fread(), メモリ, 改行, 応用全般	f 開, コメント	—	fp, f 閉, f 入出力	—
5L	17 名	ループ, コマンド行, 構造体, 標準出力, 列入出力, fp, f 開閉, f 入出力, f 終端, fread(), メモリ, 改行, コメント, 応用全般	—	—	—	—

れたため、第 2 週開始時点の値は示していない。優先度 A と判定された受講者に加え、優先度 B と判定された者も抽出する場合は、再現率は 79.3～86.2%と高い値を示すが、適合率は 39.0～42.4%に下がる。これは、抽出される受講者の中に、習得不十分者でない受講者のほうが多く含まれることを示しているが、習得不十分者でない受講者が指導を必要としない受講者であるというわけではないと考えられる。また、本当に指導すべき受講者に指導を与えられないことに比べ、優先して指導する必要がないかもしれない受講者に対して指導を与えることの悪影響は少ないと考えられる。優先して指導する必要がないとしても、すべての学習項目を理解できているとは限らないので、抽出されたことが、理解できていない学習項目の指導を行うきっかけとなりうる可能性がある。また、本当に必要なければ、教員が判断して指導を行わないことも可能である。これらのことをふまえると、適合率よりも再現率を重視するべきであるといえる。

優先度 B と判定された受講者まで含める場合でも、全 146 名から 5 分の 2 の 59 名まで絞ることができ、そのうえで高い再現率を示しているため、この基準での抽出結果を教員に提示することで、限られた時間での指導効果を高めることができると考えられる。したがっ

て、本手法により、行き詰まりのおそれが高い受講者を、教員が優先して指導すべき受講者として、早い時点から適切に抽出できるといえる。

5.3 グルーピング結果

続いて、優先指導の必要性が高いと判定された受講者を、理解状況の特徴に基づいてグルーピングできるかどうかを検証した。ここでは、受講者にとって最初の応用技法であるリスト構造を学んだ後の第 5 週開始時点と、当科目の中間にあたる第 8 週開始時点を対象に検証を行う。提案手法を実際の現場に適用する際は、現在の学習者の中で誰が習得不十分者になるかが明らかになっていない。そのため、優先して指導すべき受講者として抽出された受講者群を対象にグルーピングを行うことになる。そこで、抽出する基準を優先度 A または B と判定された受講者とする場合を想定し、その基準に該当する 59 名を対象にグルーピングを行った結果を示す。

表 6 に、第 5 週開始時点で優先度 A または B と判定された 59 名をグルーピングした結果を示す。構成されたグループごとに、該当する学習者数と、それらの学習者が他グループの学習者に比べて理解度合および取り組み度合が低い学習項目を、

表 7 表 6, 表 9 における学習項目名の略称
Table 7 Abbreviated learning item name.

略称	学習項目名
ループ	ループ (for 文/while 文)
コマンド行	コマンド行の引数
構造体	構造体の宣言/定義, および, 構造体の操作
標準出力	標準出力への出力
字入力, 字出力 (字入出力)	文字の入力, 文字の出力
列入力, 列出力 (列入出力)	文字列の入力, 文字列の出力
fp	ファイルポインタの宣言
f 開, f 閉 (f 開閉)	ファイルのオープン, ファイルのクローズ
f 入力, f 出力 (f 入出力)	ファイルからの入力, ファイルへの出力
f 終端	ファイルの終端判定
fread()	fread()/fwrite()
メモリ	メモリ領域の確保, および, メモリ領域の解放
終了時解放	終了時の全メモリ解放
改行	空白/改行文字の処理
コメント	コメントの記述
応用全般	リスト構造, スタック, キュー, 探索法に関する学習項目

- 取り組み度合が 0 もしくは 0 に近い学習項目,
- 理解度合が 0 もしくは 0 に近く, 取り組み度合も低い学習項目,
- 理解度合は 0 もしくは 0 に近いが, 取り組み度合は低い学習項目,
- 理解度合が比較的 low, 取り組み度合も低い学習項目,
- 理解度合は比較的 low, 取り組み度合は低い学習項目,

に分類して示している。学習項目名は略称で示しており, 表 7 にその対応を示す。

理解度合および取り組み度合の高低は, 全グループの平均を基準に判定した。取り組み度合が 0 もしくは 0 に近い学習項目は理解度合が未知であるが, 当該学習項目を理解できていないために, それが含まれる演習課題に取り組んでいない可能性が推測される。理解度合が低い学習項目のうち, 取り組み度合も低い学習項目は, 取り組みが少ないことが理解度合が低くなっている原因であるとも考えられ, 理解できていないとは断定しがたい。一方, 取り組み度合が低い学習項目は理解できていない可能性が高く, 指導が必要であるといえる。理解度合が 0 の場合はまったく理解できていないといえるのに対し, 比較的 low の 0 ではない場合は理解に近づいている, もしくは, 理解できた直後である可能性が考えられるので, 両者を区別している。また, 取り組み度合が低い学習項目が多いグループは, すでに行き詰まっているおそれが高いと考えられる。

表から分かるように, 12 グループが構成され, それぞれが他グループとは異なる理解状況を示している。2006 年度のどの習得不十分者の理解状況と最も類似しているかによってグルーピングを行っているため, 各グループの理解状況は 2006 年度の習得不十分者のうちの 1 名以上の理解状況を表していることとなる。よって, 各習得不十分者の当該時点での理解状況はそれぞれ異なっていることが分かる。それゆえ, 優先指導の必要性が高いと判定された受講者でも, どのグループに分類されるかによって, 指導すべき内容が異なるといえる。たとえば, 全グループともこの時点ではリスト構造の学習に取り組めていないが, グループごとに考えられる原因は異なる。5G グループは, リスト構造の実現に深く関わると思われる基礎技法の理解不足が見られない。よって, リスト構造の概念や原理を中心に指導すべきと考えられる。5A, 5B, 5C の各グループは, 取り組み度合が 0 の学習項目にメモリ管理に関する学習項目が含まれていることから, その理解不足によって学習が遅れている可能性が考えられる。それゆえ, リスト構造に関する指導の前に, メモリ管理に関する学習項目が理解できているかを確認したほうがよい。5F グループはメモリ管理単元の理解不足が明らかであり, それに対する指導が必須となる。その他のグループは, さらに構造体単元の理解についても注意が必要となる。また, リスト構造の実現には深くかわからないものの, 現時点で指導しておいたほうがよい学習項目もグループごとに明らかになっている。

ここで, 優先して指導すべき受講者として抽出された受講者群と抽出されなかった受講者群との理解状況の特徴の違いを確認するため, 全受講者 146 名を対象にグルーピングした結果について考察する。表 8 は, 第 5 週開始時点で全受講者 146 名をグルーピングした結果, 各グループに分類された受講者の優先度別の人数と, そのうちの習得不十分者数を示している。表から分かるように, 優先度 A または B と判定された 59 名のみをグルーピングした場合よりも 6 グループ増えて, 18 グループが構成された。新たに構成された 6 グループはもとの 12 グループに比べ, ほとんどの学習項目について, 取り組み度合, 理解度合ともに高い値を示しており, 5R グループを除いては, もとの 12 グループが取り組めていないリスト構造に関する学習項目にも取り組んでいる。優先度 C~E と判定された 87 名のうち, 約 4 分の 3 にあたる 64 名が, 優先度 A または B と判定された者が属していない 5M~5R の 6 グループのいずれかに分類されている。優先度 A または B の受講者が属する 5A~5L の 12 グループのうち, 優先度 C の受講者も属しているグループは 4 グループしかなく, 優先度 D または E の受講者が属しているグループは 5A グループのみである。これより, 優先して指導すべき受講者として抽出された優先度 A または B の受講者群と, 優先度 C~E の受講者群とは理解状況が大きく異なっていることが分かる。また, この時点で

表 8 第 5 週開始時点での全受講者を対象としたグルーピング結果
Table 8 Grouping results of all students at the beginning of 5th week.

グループ	優先度 A	優先度 B	優先度 C	優先度 D	優先度 E	合計
5A	0 名	2 名(0 名)	6 名(2 名)	5 名(2 名)	1 名(0 名)	14 名(4 名)
5B	0 名	7 名(2 名)	6 名(0 名)	0 名	0 名	13 名(2 名)
5C	0 名	3 名(1 名)	4 名(0 名)	0 名	0 名	7 名(1 名)
5D	5 名(0 名)	3 名(0 名)	0 名	0 名	0 名	8 名(0 名)
5E	0 名	8 名(2 名)	0 名	0 名	0 名	8 名(2 名)
5F	0 名	2 名(1 名)	0 名	0 名	0 名	2 名(1 名)
5G	0 名	1 名(0 名)	1 名(0 名)	0 名	0 名	2 名(0 名)
5H	3 名(1 名)	0 名	0 名	0 名	0 名	3 名(1 名)
5I	2 名(1 名)	2 名(1 名)	0 名	0 名	0 名	4 名(2 名)
5J	0 名	1 名(0 名)	0 名	0 名	0 名	1 名(0 名)
5K	3 名(0 名)	0 名	0 名	0 名	0 名	3 名(0 名)
5L	17 名(15 名)	0 名	0 名	0 名	0 名	17 名(15 名)
5M	0 名	0 名	0 名	2 名(0 名)	0 名	2 名(0 名)
5N	0 名	0 名	2 名(0 名)	1 名(0 名)	0 名	3 名(0 名)
5O	0 名	0 名	1 名(0 名)	0 名	0 名	1 名(0 名)
5P	0 名	0 名	2 名(0 名)	16 名(0 名)	28 名(0 名)	46 名(0 名)
5Q	0 名	0 名	1 名(0 名)	0 名	0 名	1 名(0 名)
5R	0 名	0 名	6 名(0 名)	5 名(1 名)	0 名	11 名(1 名)

() 内は習得不十分者数を表す。

は優先度 C~E と判定された習得不十分者 5 名のうち、4 名が 5A グループ、1 名が 5R グループに分類された。これより、現時点では優先度が低くても、優先度が高い者が属しているグループに分類されている場合は注意が必要であるといえる。これは、優先度 A または B の受講者が属している 12 グループのうち、8 グループには習得不十分者が含まれていることから分かる。それゆえ、現時点では優先度が低くても、優先度が高い受講者と同じグループに分類されている受講者群を、次に指導すべき受講者群として教員に提示すれば、行き詰まりをより防げると考えられる。

提案手法では、指導を必要とする受講者のグルーピングを目的とし、過去の習得不十分者のデータに基づいているため、優先度が低い受講者を細かくグルーピングすることは難しい。そのため、優先度 D または E の受講者の多くが 5P グループに集中している。しかし、優先度が低い受講者の多くが、優先度が高い受講者とは異なるグループに分類されており、優先度が高い受講者については理解状況に基づいて細かくグルーピングを行っている。したがって、プログラミング学習カルテでは、詳細な理解状況を的確に表せているといえる。

表 9 に、第 8 週開始時点で優先度 A または優先度 B と判定された 59 名をグルーピング

した結果を表 6 と同様に示す。グループ名は順不同であり、グループ名に含まれるアルファベットが同じであっても、第 5 週開始時点でのグループとの関連はない。この時点でも 12 グループが構成されたが、先のグルーピング結果とは構成が異なり、各グループの特徴も異なる。これより、第 5 週開始時点から 3 週間の学習経過も反映したグルーピングを行っていることが分かる。また、この時点においても、すべてのグループが他グループとは異なる特徴を示している。したがって、指導優先度が高いと判定された受講者群を、理解状況が似た学習者ごとにグルーピングできたといえる。さらに、プログラミング学習カルテから明らかになる、各グループの学習項目単位での詳細な特徴情報により、教員は指導指針を立てやすくなると考えられる。

第 5 週開始時点の 5L グループや第 8 週開始時点の 8L グループのように、すべての学習項目が理解度合、取り組み度合ともに 0 もしくは 0 に近いグループがある一方で、5B グループや 8H グループは、他のグループに比べて理解度合および取り組み度合の低い学習項目が少ない。しかし、2006 年度、2007 年度ともに、これらのグループに分類され、かつ、結果的に習得不十分者となった受講者が存在する。これより、すべての学習項目の理解度合および取り組み度合が 0 もしくは 0 に近い受講者のみが習得不十分者になっているわけではなく、前半では順調に学習を進めている受講者でも習得不十分者になりうるということが分かる。そして、そのような受講者も、過去の学習者のデータを用いることによって早い段階で同定できることが、今回の検証で確認された。したがって、優先して指導すべき学習者を抽出するうえで過去の学習者のデータは有用であるといえる。

このように、本手法を大学の C 言語プログラミング科目に適用した結果、本手法は行き詰まりのおそれが高い、教員が優先して指導すべき受講者の抽出とグルーピングを、早い時点から適切に行えることが分かった。したがって、本手法は多数の学習者に対して限られた時間でより充実した指導を行うための情報を教員に提供できる。ゆえに、本手法は、プログラミング教育における限られた人的教育資源の有効活用を実現するといえる。

5.4 考 察

実データを用いた検証により、提案する手法が、3.1 節で述べた 3 種類の教員の負荷のうち、指導負荷を軽減させるために有用な指導支援情報を提供できることが確認された。ここでは、本手法を実際の教育現場に適用するうえでの、残りの 2 つの負荷について考察する。考察にあたり、検証を行った演習科目を担当した教員 4 名、および、異なる年度に同科目を担当した教員 1 名の計 5 名の教員と、当科目で TA を担当した大学院生 9 名に対して調査を実施した。これらの 14 名に対して当科目での指導に関する質問を行い、回答を得た。教

表 9 第 8 週開始時点におけるグルーピング結果
Table 9 Grouping results at the beginning of 8th week.

グループ	人数	取り組み度合が 0 もしくは 0 に近い	理解度合が 0 もしくは 0 に近い		理解度合が低い	
			取り組み度合が低い	取り組み度合が低くない	取り組み度合が低い	取り組み度合が低くない
8A	1 名	構造体, 標準出力, fread(), メモリ, 終了時解放, 応用全般	—	—	字入出力, 改行	—
8B	11 名	構造体, fread(), メモリ, 終了時解放, 応用全般	字入力, 改行	—	字出力	f 開
8C	9 名	メモリ, 終了時解放, 応用全般	字入出力	—	f 出力	コメント
8D	3 名	応用全般	—	メモリ, 終了時解放	改行	字入出力, コメント
8E	2 名	コマンド行, 標準出力, メモリ, 終了時解放, 応用全般	—	—	ループ, 字入出力, 列入出力, fp, f 開閉, f 入力, f 終端, 改行	—
8F	1 名	コマンド行, 構造体, 標準出力, fread(), メモリ, 終了時解放, 応用全般	—	—	ループ, 字入出力, 列入出力, fp, f 開閉, f 入力, f 終端, 改行	—
8G	1 名	コマンド行, 構造体, 標準出力, fread(), メモリ, 終了時解放, 応用全般	コメント	—	ループ, 字入出力, 列入出力, fp, f 開閉, f 入力, f 終端, 改行	—
8H	13 名	終了時解放, 応用全般	—	—	—	—
8I	1 名	ループ, コマンド行, 構造体, 標準出力, 列入出力, f 終端, fread(), メモリ, 終了時解放, 改行, 応用全般	コメント	—	字入出力, fp, f 開閉, f 入出力	—
8J	1 名	コマンド, 標準出力, 終了時解放, 単リストを除く応用全般	—	—	ループ, 字入出力, 列入出力, fp, f 入力, f 終端, 改行	—
8K	2 名	メモリ, 終了時解放, 二分探索を除く応用全般	—	—	—	コメント
8L	14 名	ループ, コマンド行, 構造体, 標準出力, 字入出力, 列入出力, fp, f 開閉, f 入出力, f 終端, fread(), メモリ, 終了時解放, 改行, コメント, 応用全般	—	—	—	—

員 5 名にはさらに、本手法の効果についての質問も行った。

まず、評価負荷について考察する。本手法の実現のためには、学習者の演習課題の評価結果を確実に得ることが不可欠となる。本手法が求める評価結果は、演習課題で実現する内容を細かく区切った評価観点ごとに、実現できているか、実現できていないかの選択式で判定した結果である。このような客観的かつ明確な評価方法であれば、評価者が異なっても統一基準での評価結果を得ることができる。よって、教員 1 人ですべての評価作業を行うことが必須ではなくなり、すでにプログラミングを習得している上回生などを評価作業員として配置して作業を分担することが可能になる。さらに、我々は評価作業をサポートするプログラミング学習カルテシステムを提供する。教員および評価作業員が当システム上で評価作業を行えば、学習項目ごとの取り組み度合と理解度合が算出できるようになり、本手法に基づく指導支援情報を得ることができる。そのうえ、学習者には評価結果が提示されるので、学習者の学習意欲を高められる。教員に対する調査結果からは、本手法が求める評価方法

の負荷について、「評価作業員のレベルおよび人数を確保できれば大きな効果が得られるが、評価作業員が少ないと負荷が大きくなりすぎる」との意見を多く得た。この評価方法によって得られた効果としては、「多くの教員および TA で評価の基準を揃えることができる」、「受講者にとっても、評価結果を得ることで、自身のソースコードがどの点で良いか悪いが、何に気を付ければよいのかがはっきりする」といった点があげられた。したがって、評価作業員の確保が可能であれば、教員の評価負荷を増大させることなく、評価作業の成果を学習者の能力向上のために大いに生かすことができるといえる。なお、当科目では、教員と TA がそれぞれ受講者約 7 名分の評価を担当し、提出されたすべてのソースコードを評価した。

次に、課題作成負荷について考察する。本手法では、演習課題の評価観点は課題を作成する教員があらかじめ定めることを想定している。そのため、演習課題を作成する際に、評価観点の作成と学習項目の対応付けの作業が追加される。しかし、作成する演習課題および評価観点の質が高いほど、上述のとおり評価負荷と指導負荷を軽減させることが可能とな

る。さらに、質が高い演習課題および評価観点を1度作成すれば、翌年度以降も再利用できる。したがって、本手法を適用する最初の段階では課題作成負荷は増大するが、その他の負荷と翌年度以降の課題作成負荷の軽減につながるため、実質的には負荷の増大は少ないといえる。また、本手法によるそれらの効果が、教員の課題作成に対するモチベーションを高め、良質の演習課題が作成されるといった良い影響を生むことも期待される。これらの点を確認するため、演習課題および評価観点的作成を担当した教員に「演習課題と評価観点的作成にかかった負荷以上の効果は得られたか」という質問を行ったところ、全員から「課題作成負荷以上の効果が得られた」との回答を得た。得られた効果として、「課題作成者が課題の意図や評価の際に重視すべきポイントを明らかにでき、作成者以外の者が課題に関する質問への回答や指導を的確に行える」ことがあげられた。また、評価観点的作成に際し、「適切なコメントを記述できているかなど、単純に実現できているかどうかで判定しにくい内容の扱いに困る」、「設定されている評価観点から外れる成果があった場合に評価しにくい」、「評価観点的の粒度や判定の難しさによっては、受講者が評価結果を見たときに逆に混乱してしまうおそれがある」との意見を得た。したがって、評価観点的の作成方法については議論の余地が残されているが、本手法にともなう課題作成負荷を超える効果は得られるといえる。なお、当科目では、各クラスを担当する4名の教員が分担して演習課題および評価観点的の作成にあたり、全59題の演習課題を作成した。

ここまでの考察から、今回の検証では指導支援情報を提供していないにもかかわらず、本手法が求める演習課題の評価方法によって、それ以外の効果が得られていることが分かる。続いて、本手法による指導支援情報を提供した場合の効果について、「本手法が提供する情報によって指導の質を向上させることができるか」という質問を行った。これに対して、4名から「大きく向上させられる」、1名から「向上させられる」との回答を得た。向上させられる理由として、

- 指導すべき学習者が判明し、その学習者がどこまで理解しているかが分かれば、1人あたりの指導時間を短縮でき、全体としての効率を上げられる、
- 多人数の学習者に対して、個別の理解状況を把握することや、適宜統一した基準で指導を優先すべき学習者を抽出することは困難であるから、それが可能になるならば指導の質を向上させられる、
- 学習者が個別に演習をしている状態では個々の到達度を把握しにくいので、到達度が可視化されることで、指導の質を向上させることは十分可能である。ただし、プログラミング演習に対する学習者の動機付けが低い場合には、指導の質を向上させても学習効果

は必ずしも上がらない可能性はある、

- 経験上、指導が必要な学習者は教員へあまり質問せず、教員側が様子を察知して聞き出すことで初めて分からない点を伝える。それゆえ、優先して指導すべき学習者が提示されると、教室内を見回って様子を察知する時間が減る分だけ指導時間を長く確保できるので、指導の質を大きく向上させることができると考える。また、学習カルテから全体の理解状況を把握することによって、課題解説などの指導内容を調整できる点でも指導の質は大きく向上すると考える、

といった点があげられた。また、「同じような理解状況の学習者をまとめて指導できれば、学習者全体の能力向上を図るうえで効果的か」という質問に対しては、4名から「大きな効果が得られる」、1名から「効果が得られる」との回答を得た。その理由として、

- 演習形式の授業では到達度に差が出るのは当然であり、個々の学習者に合わせた指導を行うことは不可欠であるが、到達度が似ている学習者でグルーピングを行うことができれば、限られた教員/TA数で個別指導を行うことができる、
- 同じことを何度も言わなくてよいので時間を効率的に使える、
- 自分と同じ間違いをしている者が多いと気付くと、学習者は質問がしやくすなる、
- 学習者同士で理解できた部分やできなかった部分などを相談するようになり、他の学習者の理解の仕方を見れば共感やモチベーションを覚える、
- 指導する内容によるが、データ構造やアルゴリズムの概念を把握させるためであれば、まとめて指導することの効果大きい。ただし、ソースコードの書き方に関連する指導の場合は、書き方が異なる学習者をまとめて指導しても効果は小さいと感じる、

といった点があげられた。今回の検証では、第5週開始時点、第8週開始時点のどちらにおいても、優先して指導すべき受講者として抽出された59名が12グループにグルーピングされた。1名ずつ指導した場合と、グループ単位でまとめて指導した場合とでかかる時間や負荷を単純に比較することはできないが、両方の場合において同様の指導内容を与えれば、後者は前者に比べて指導回数を約5分の1に減らすことができる。グループ単位での指導によって、同じ指導を繰り返すことにかかる時間と負荷を削減でき、より多くの学習者に対して指導を与えることが可能になるうえに、学習者のモチベーションを上げることも期待される。したがって、本手法が提供する指導支援情報は、教員が効率的かつ効果的に指導を行うために十分に有用であるといえる。

さらに、当科目で教員やTAが行った指導内容についても、調査結果に基づいて考察した。どのような方法で指導することが多かったか。授業時間中の対面指導が突出して多く、次い

で、学習カルテシステムの評価コメント欄に記述する形での指導、授業時間外の対面指導という順で多かった。この結果より、プログラミング教育において、授業時間中の対面指導は非常に重要であり、教員に限られた時間で、指導を必要とする学習者にはできるだけ対面指導を行えるようにすることが、教育効果を高めるうえで有効であることが改めて確認されたといえる。

プッシュ型指導とプル型指導のどちらを多く行ったか 教える側から学習者に話しかけて指導するプッシュ型指導のほうが多かったと回答したのが6名、学習者側から質問があれば対応するプル型指導のほうが多かったと回答したのが8名であった。プル型指導のほうが上回る結果となったが、プッシュ型指導との回答も少なくなかったことから、指導を必要としている学習者が必ずしも自ら積極的に教員に指導を求めないことが分かる。したがって、優先して指導すべき学習者を抽出して教員に提示し、プッシュ型指導を促進させることは効果的であるといえる。

どのような内容の指導を多く行ったか 課題内容の解説やヒントの提示が最も多く、次いで、学習者が作成したソースコードの修正やデバッグ、ソースコードの良くない点の指摘が多くあげられた。そのほかには、基礎技法についての説明、講義科目の講義資料を用いた解説、課題の解答となるソースコードの一部を提示などがあげられた。

行った指導のうち効果が大きかったものはどれか 課題内容の解説やヒントの提示、および、学習者が作成したソースコードの良くない点の指摘が最も多く、次いで、基礎技法についての説明という結果であった。ソースコードの修正やデバッグはよく行われていたものの、その効果はあまり大きくなかったようである。課題内容の解説やヒントの提示、基礎技法についての説明は、同じような指導が必要な学習者をまとめて1度に指導することが可能である。また、ソースコードの良くない点の指摘は、理解状況が似た学習者グループに対して行うことで、ある学習者のソースコードへの指摘が、他の学習者の理解の促進や気付きにつながる効果が期待される。

これらの考察より、本手法を適用し、行き詰まりのおそれが高い学習者を優先的に、理解状況が似た学習者をまとめて指導すれば、現状よりも充実した指導を実現できるといえる。

ゆえに、本手法は、教員の課題作成負荷および評価負荷を著しく増大させることなく、効率的な指導を実現するために教育現場に適用できるといえる。さらに副次的な効果も与えることから、限られた時間での教育効果を高められるといえる。

6. 関連研究

本論文で述べた内容の成果は、個々の学習内容が複雑に絡み合うプログラミング教育において、学習者の理解状況を的確かつ容易に把握するための仕組みと、それをを用いて教員に効率的な指導のための支援情報を与える手法を提供したことである。

プログラミング言語の初歩的な文法を自習するための学習環境⁵⁾や、プログラミング初級者の達成度をテストする手法⁶⁾など、プログラミング教育支援を目的に提案されている手法やツールには、基礎技法の教育を対象としたものが多い。文献7)や文献8)も、同じく基礎技法のプログラミング演習において、コンパイルエラーの内容に応じた支援情報を学習者に提供する手法を提案している。応用技法の教育では、基礎技法に比べて学習者の行き詰まりの原因も複雑になるため、そのような局所的なデータのみでは適切な支援法を決めることが困難である。本手法では、ソースコードを細かい評価結果に基づいて学習者の理解状況を明らかにすることで、応用技法の教育を主に支援する。

proGrep⁹⁾やIPP¹⁰⁾は、プログラミング学習者の学習履歴を収集して活用するためのシステムである。proGrepはプログラミング演習における学習履歴を収集して、教員がそれらを検索できる機能を提供をしているが、収集するデータは先に述べた手法と同様に、コンパイルや実行の結果である。IPPはプログラミング課題の成果物を一連の記録として閲覧できる機能を提供しているが、学習者の理解状況を表すデータは収集していない。また、対象を特に限定せず、電子教材を用いた学習での学習履歴を視覚化するシステム^{11),12)}も提案されている。これらのシステムでも学習の過程を確認することはできるが、理解状況は容易に把握できない。

文献13)では、英語教育を対象に、個々の学習者の理解状況を教員に提示する手法が提案されている。提示される情報は、細分化されたスキルごとの小テストにおける正解率である。これに対して本手法では、取り組み度合と理解度合の2つの指標を用いて、すべての学習者の理解状況を統一した基準で表現している。

学習者の認知構造を表現した認知マップを用いた手法¹⁴⁾では、学習内容のまとまりごとに学習者の理解状況が評価される。まとまりごとの理解状況は定量的に求められるが、含まれる細目ごとの理解状況は明らかにならず、本手法のような理解状況の分析が行えない。

行き詰まっている学習者や異常な学習プロセスを示している学習者を検出する手法も提案されている¹⁵⁾⁻¹⁷⁾。これらの手法では、多肢選択問題や空欄補充問題といった学習コンテンツに対する学習所要時間のデータを用いて検出を行っている。応用技法のプログラミング課

題では1題あたりの所要時間が長く、理解状況が所要時間に反映されるとは限らないので、これらの手法を適用することは難しい。また、検出された学習者に対する教員の指導支援としては、たかだか学習進度の提示にとどまっている。本手法では、行き詰まりのおそれが高い学習者を抽出するだけでなく、それらの学習者を理解状況に応じてグルーピングした結果も教員に提示する。

7. おわりに

本論文では、プログラミング教育において、教員が学習者の理解状況を把握するためのプログラミング学習カルテの実現法と、その分析によって教員が限られた時間で教育効果を高めるための指導支援情報を提供する手法を提案した。プログラミング学習カルテは、プログラミング演習課題の評価結果に基づいて構成され、教員はつねに最新の理解状況を把握することができる。過去の学習者のデータを用いてプログラミング学習カルテを分析することで、優先して指導すべき学習者の抽出と、抽出された学習者の理解状況に応じたグルーピングを行う。それらの結果の提供により、教員は指導にかかる負荷を軽減でき、限られた人的教育資源でより充実した指導を実現できる。

本手法に基づいてプログラミング学習カルテを実現するシステムを構築し、大学のC言語プログラミング科目に適用した。その結果、本手法を用いることにより、早い時点で行き詰まりのおそれが高い学習者の抽出とグルーピングを適切に行うことができた。

本手法は、教員の負荷のうち、指導にかかる負荷の軽減を目的としている。本手法の適用によってその他の負荷を著しく増大させることはないが、課題作成作業や評価作業を直接支援する手法やツールを提供すれば、それらに必要な人的教育資源を指導に用いることができ、本手法の有効性がさらに高まる。そのような手法やツールの実現は今後の課題である。

参考文献

- 1) 日本経済団体連合会：産学官連携による高度な情報通信人材の育成強化に向けて (2005). <http://www.keidanren.or.jp/japanese/policy/2005/039/index.html>
- 2) Kernighan, B.W. and Ritchie, D.M.: *The C Programming Language*, 2nd edition, Prentice Hall, Englewood Cliffs (1988). 石田晴久 (訳): *プログラミング言語C 第2版*, 共立出版 (1989).
- 3) 今泉貴史: *プログラミングに活かすデータ構造とアルゴリズムの基礎知識*, アスキー, 東京 (2004).
- 4) 田口 浩, 糸賀裕弥, 山本哲男, 高田秀志, 島川博光: *プログラミング演習評価と講*

義反応を連携させた理解の契機の抽出, 電子情報通信学会論文誌 D, Vol.J91-D, No.2, pp.345-357 (2008).

- 5) Hulls, C.W.C., Neale, J.A., Komalo, N.B., Petrov, V. and Brush, J.D.: Interactive Online Tutorial Assistance for a First Programming Course, *IEEE Trans. Education*, Vol.48, No.4, pp.719-728 (2005).
- 6) Farrow, M. and King, J.B.P.: Experiences with Online Programming Examinations, *IEEE Trans. Education*, Vol.51, No.2, pp.251-255 (2008).
- 7) 藤原祥隆, 松西年春, 岡田信一郎, 大鎌 広, 後藤寛幸, 黒丸鉄男: *プログラミング演習支援のための階層分散処理システムの設計と評価*, 電子情報通信学会論文誌 D-II, Vol.J78-D-II, No.11, pp.1701-1709 (1995).
- 8) 高本明美, 藤井美知子, 泉 直利, 田中 稔: 誤り原因を指摘するプログラム学習支援システムとその学習効果, *教育システム情報学会誌*, Vol.17, No.4, pp.533-540 (2001).
- 9) 長 慎也, 箕 捷彦: proGrep—プログラミング学習履歴検索システム, *情報処理学会研究報告*, No.2005-CE-78, pp.29-36 (2005).
- 10) Estell, K.J.: IPP: A Web-Based Interactive Programming Portfolio, *ACM SIGCSE Bulletin*, Vol.33, No.1, pp.149-153 (2001).
- 11) 大川正人, 室田真男, 中山 実, 清水康敬: Web ベース学習における学習履歴画面の時系列再現システムの開発, 電子情報通信学会論文誌 D-I, Vol.J83-D-I, No.6, pp.651-657 (2000).
- 12) 金西計英, 妻鳥貴彦, 矢野米雄: LOGEMON: Web 教材を使用した授業での教師支援システム—学習者の閲覧履歴の視覚化による教師支援, 電子情報通信学会論文誌 D-I, Vol.J83-D-I, No.6, pp.658-670 (2000).
- 13) 高橋 幸, 小林昌博, 安浪誠祐, 中野裕司: 学習者カルテを利用した英語教育支援システムの構築, 電子情報通信学会技術研究報告, No.TL2004-24, pp.29-33 (2004).
- 14) 竹谷 誠, 佐々木整: 学習者描画の認知マップによる理解度評価法, 電子情報通信学会論文誌 D-II, Vol.J80-D-II, No.1, pp.336-347 (1997).
- 15) 桑原恒夫, 玉城幹介, 山田光一, 中村喜宏, 満永 豊, 小西納子, 天野和哉: 個人進捗別教育支援システム (MESIA) における行き詰まり生徒の支援機能とその効果, 電子情報通信学会論文誌 D-I, Vol.J83-D-I, No.9, pp.1013-1024 (2000).
- 16) 中村喜宏, 赤松則男, 桑原恒夫, 玉城幹介: 操作時間間隔の変動に着目した CAI 学習の行き詰まり検知方法, 電子情報通信学会論文誌 D-I, Vol.J85-D-I, No.1, pp.79-90 (2002).
- 17) 植野真臣: e ラーニングにおける所要時間データの異常値オンライン検出, 電子情報通信学会論文誌 D, Vol.J90-D, No.1, pp.40-51 (2007).

(平成 20 年 11 月 26 日受付)

(平成 21 年 7 月 2 日採録)



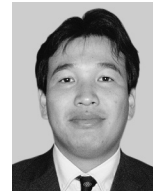
田口 浩 (正会員)

昭和 57 年生。平成 16 年立命館大学工学部情報学科卒業。平成 18 年同大学大学院理工学研究科博士前期課程修了。平成 21 年同大学院理工学研究科博士後期課程修了。同年三菱電機(株)入社。博士(工学)。現在、同社先端技術総合研究所にてエレベータシステム等の研究開発に従事。電子情報通信学会会員。



原田 史子 (正会員)

昭和 55 年生。平成 15 年大阪大学基礎工学部システム学科卒業。平成 16 年同大学大学院基礎工学研究科博士前期課程修了。平成 19 年同大学院基礎工学研究科博士後期課程修了。同年立命館大学情報理工学部情報システム学科助教。平成 21 年より同大学情報理工学部情報システム学科講師。博士(工学)。リアルタイムシステム、データ工学の研究に従事。電子情報通信学会、システム制御情報学会、IEEE 各会員。



高田 秀志 (正会員)

昭和 43 年生。平成 3 年京都大学工学部情報工学科卒業。平成 5 年同大学大学院工学研究科情報工学専攻修士課程修了。同年三菱電機(株)入社。平成 13 年京都大学大学院情報学研究科社会情報学専攻博士後期課程修了。平成 16 年京都大学情報学研究科 COE 研究員。平成 18 年より立命館大学情報理工学部情報システム学科准教授。博士(情報学)。分散システム、協調システム、教育・学習システム等の研究に従事。システム制御情報学会、ACM 各会員。



島川 博光 (正会員)

昭和 36 年生。昭和 59 年京都大学工学部情報工学科卒業。昭和 61 年同大学大学院工学研究科博士前期課程修了。同年三菱電機(株)入社。平成 11 年京都大学大学院工学研究科博士後期課程修了。平成 14 年立命館大学理工学部情報学科教授。平成 16 年より同大学情報理工学部情報システム学科教授。博士(工学)。データベース、プログラミング教育、ユビキタス環境の研究に従事。電子情報通信学会、システム制御情報学会、ACM、IEEE 各会員。