

LK-003

プログラミング初学者のための学習モデル RPRaS の提案

Proposal for a Learning Model "RPRaS" for Novice Programming

水谷 晃 三†
Kozo Mizutani赤羽根 隆広†
Takahiro Akabane荒井 正 之†
Masayuki Arai海上 隆†
Takashi Unagami

1. まえがき

本論文では、プログラミング初学者のための新しい学習モデルとして RPRaS (Relation Between Program Requirements and Program Sources) を提案する。さらに、本モデルに基づいた学習支援の準備として、試作システムを用いた基礎研究結果を報告する。

プログラミング学習を支援するための多くの理論、モデル及びシステムが数多く提案されている。筆者らも同様に、プログラミング初学者を対象とした研究を行っている。筆者らは、プログラミング初学者が次の4つの過程を経てプログラミング能力を習得するものと考えている。

- (1) プログラムの基礎と全体構成の理解
- (2) アルゴリズムの基本構成単位である接続・判断・反復の理解と、それらを使ったプログラムをトレースする能力の習得
- (3) アルゴリズムの基本構成単位を用いて、プログラムを作成する能力の修得
- (4) 各種アルゴリズムやデータ構造をプログラムとして記述する能力の習得

(2) については、筆者らは Web ベースの学習支援システムを既に提案している[1]。同様に、プログラムソースとプログラムの振舞いの理解を促進するために、いくつかの学習支援モデルやシステムが提案されている。近年では、例えば寺田らによるトレースビューアがある[2]。Milne らはプログラム実行中のメモリの状態を視覚化するツールを提案している[3]。

(3) については開発支援環境 (IDE) を活用することができる。しかし、IDE は大規模なソフトウェア開発を想定したツールであることが多く、初学者にとって適切とはいえない場合がある。その様な観点から、初学者プログラマに関する研究や支援システムが提案されている[4-8]。

以上の方法によって、プログラムソースを書き、その動作を確かめるといふ過程の能力を向上させることは可能であると考えられる。しかし、プログラム要件とプログラムソースの関連を理解する能力が同時に向上するとは考えにくい。実際、プログラム要件を満たすプログラムを作成できない学生を、プログラミング初学者を対象とした授業において確認することがある。そのような学生は、与えられた課題が事前に例示されたプログラムに類似した要件である場合、サンプルコードを改造してプログラムを書くことは可能である場合が多い。すなわち、プログラム要件を理解していても、その要件を満たすプログラムを無の状態から書上げる能力が不足しているものと考えられる。(4) の能力を育成するために、本問題点に関する支援が必要である。

そこで本論文では、プログラム要件とプログラムソースの関連を理解するための学習モデルとして RPRaS を提案

する。RPRaS では、プログラム要件を“入力処理”、“計算などの処理”及び“出力処理”の3つに分類する。3つに分類することによって、プログラム要件とプログラムソースの関係が容易に理解できるようになると考えている。

RPRaS に基づく学習の効果をさらに高めるためには、学習支援方法の検討が必要である。そのためには事前の調査が必要である。具体的には、プログラム要件を3つに分類することの影響、効果的な学習順序、重点的に学習支援が必要とされる箇所などを調べておく必要がある。

そこで筆者らは RPRaS に基づく学習支援のための試作システムを構築した。本論文では試作システムの説明と導入結果についても述べる。

2. 学習モデル "RPRaS"

RPRaS は構造化プログラミングに基づく学習モデルである。既知のとおり、構造化プログラミングは重要なプログラミングパラダイムの1つであり多くのプログラミング言語に影響を与えている。

構造化プログラミングに関するシステム構築手法としてボックス構造化手法がある[9-11]。ボックス構造化手法では、あらゆるシステムあるいはサブシステムの振舞いは、3つの異なるボックス構造形式で表すことができると述べられている。各ボックスはいくつかの刺激 S 、反応 R 及びシステム (あるいはサブシステム) そのものを表現し、抽象度の違いからブラックボックス、状態マシン、クリアボックスと呼ばれている。

プログラミング初学者を支援するために筆者らが注目したのはボックスの形式である。ボックスは刺激 S 、反応 R 及びシステムの3つの部品で構成される。一般に、部品の役割はプログラム要件に基づいて決定される。各部品の役割を別々に定義することができるため、プログラム要件の整理が容易になりプログラミング初学者が理解しやすくなると考えている。

図1に RPRaS のモデル概念図を示す。RPRaS では、まずボックス構造の刺激 S を“入力処理”、反応 R を“出力処理”、システムを“計算などの処理”と呼ぶようにする。改名の理由は、ボックス構造化手法が大規模なソフトウェア開発をも想定した手法であるのに対し、RPRaS はプログラミング初学者を対象とする小さなプログラムを想定しているためである。

学習者はプログラム要件が与えられたとき、その要件を入力処理要件、計算などの処理要件、出力処理要件のいずれかに分類する。分類後は、要件ごとにいくつかの工程を経てから、その要件に対応するソースコードの記述を行う。各工程において学習支援を行い、効率的なプログラムソースの作成を促す。

ほとんどの場合、各工程は要件定義工程、設計工程、実装工程である。しかしながら、例えば“30ピクセルの円を描画する”という要件では、Javaなどのプログラミング言

† 帝京大学大学院理工学研究科

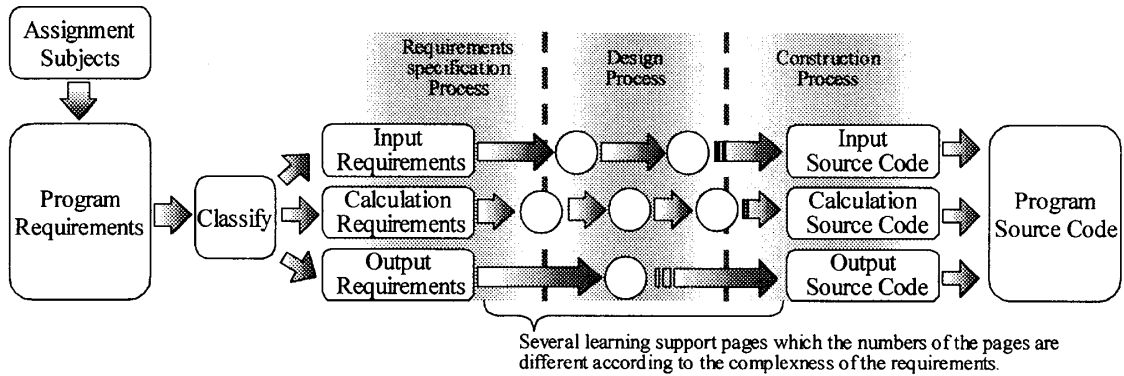


図1 学習モデル RPRaS 概要図

語では API を適切に利用するだけであるので、設計工程は不要である。逆に、複雑な要件の場合には理解が容易になるように工程の数を増やす。

学習支援の流れやタイミングが構造化プログラミング手法と一致しているので、学習者は容易にプログラム要件とプログラムソースの関係を理解することができると考えられる。3 つに分類したことで、プログラム要件全てを一度に考えなくて済むという利点もある。さらには、構造化プログラミングの手法を学習者が自然に理解することを期待できる。

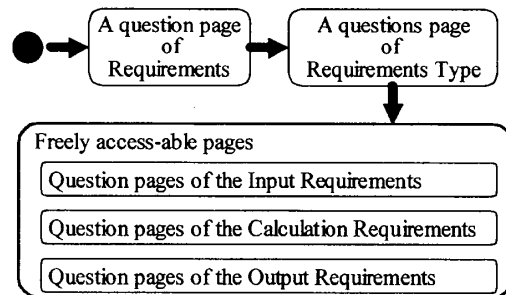


図2 試作システムの画面遷移

3. RPRaS に基づく学習システムの試作

3.1 試作システムの概要

試作システムは、Java 5, Tomcat 5.5, Microsoft SQL Server 2005, O/R マッピングフレームワークとして Hibernate 3 を用いて Web アプリケーションとして実装した。

本システムを利用するために、教授者は 1 つの課題につき、課題問題文、模範解答のプログラムソース、プログラム要件、各要件の工程ごとの学習支援ページを教材として用意する。プログラム要件は、課題問題文から教授者の判断で定義する。

これらの教材を試作システムのデータベースに登録する。登録の際、プログラム要件のタイプ（入力処理、計算などの処理、出力処理）を指定する。登録により学習者が利用可能な Web ページが自動的に生成される。学習者は 3.2 に後述する画面遷移にしたがって、学習支援を受けることができる。

3.2 試作システムの画面の実装

図 2 に試作システムの画面遷移図を示す。画面遷移は RPRaS に基づいて設計している。

まず、学習者は課題問題文を確認した後、適切なプログラム要件を選択する問題ページに回答する。次に、選択したプログラム要件のタイプを選択する問題ページに回答する。ここまでは、プログラム要件を再確認させる目的のため、学習者は画面遷移に従って学習を進める。

要件のタイプを回答すると、以降の画面はプログラム要件ごとに自由にアクセス可能な問題ページとなる。各問題ページは工程ごとの学習支援に相当する。問題ページはプログラム要件のタイプ別に画面構成しており、学習者が必要としているページにアクセスできるようになっている。

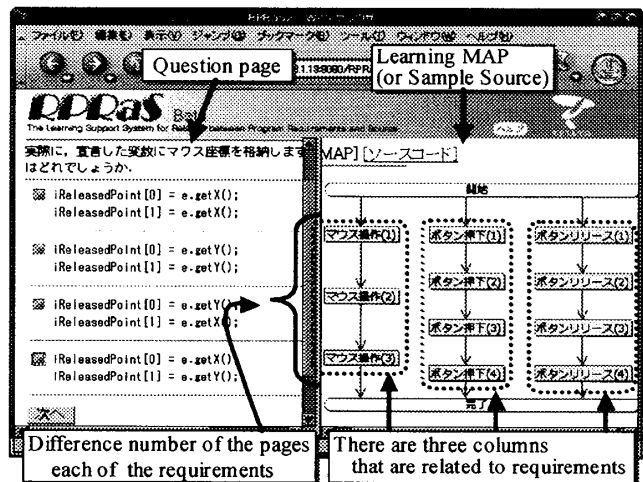


図3 試作システムの画面例

図 3 に試作システムの画面例を示す。画面左側には問題ページが表示されている。画面右側には学習マップを表示している。本マップは、問題ページの自由アクセスを案内するために導入した。本マップは RPRaS そのものであり、要件ごとに問題ページへのリンクが並べられている。

また、学習者は模範解答のサンプルソースにアクセスできるようにした。各行をクリックするとそれに対応するプログラム要件の問題ページに直接アクセスできる。ただし、問題ページを 1 つも答えてない状況では図 4 のようにモザイク処理を施す。問題ページの内容は、解答を順次進めていくにつれてソースコードそのものに近づく。これに同期するように、ソースコードの表示画面では、現在の学習進捗に対応するソースコードのモザイク処理を次第に取除く処理を実装している。この機能により、プログラム要件とソースコードの関係を視覚的に把握できるようにしている。

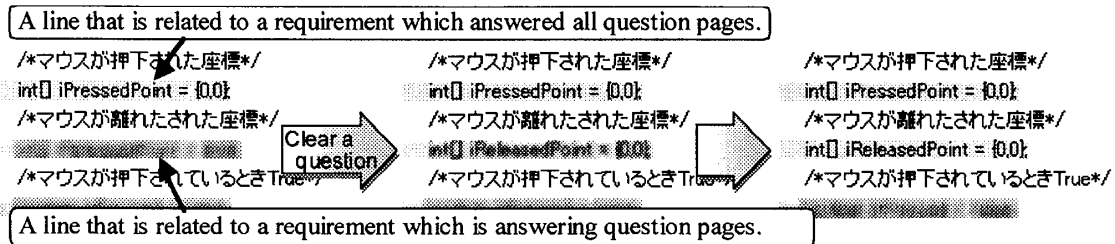


図 4 学習進捗とソースコードの表示処理の連携

4. 実験と評価

実際の授業へ導入して試作システムの評価を行った。本論文では、学習者の学習状況とレポート評価の関係を中心に調査した。

本研究で対象とする授業は、情報科学を学ぶ大学 1 年生を対象とした Java プログラミング演習である。ほとんどの受講生はプログラミング初学者であり、本授業の前に約 20 時間のプログラミング演習を受講している。

本授業では計 12 回の演習講義の中で毎回課題出題を行っているが、後半からレポート（プログラムソース）の提出率や平均評点が低下しつつあった。そこで、最後の 2 回の講義へ本試作システムを導入した。

4.1 実際の課題内容と問題ページの例

本節では実際の課題内容とそれに対応する試作システム上の問題ページでの出題例を示す。

図 5 に例示するように、課題は Java アプレットを作成する演習である。main 関数だけの一般的な初学者向け課題と違い、本課題で作成するアプレットはイベント駆動型プログラムである。そのため、要件を“入力処理”“計算などの処理”“出力処理”に分類することがやや難しい。しかし、アプレット全体を 1 つのプログラムとして捉えればイベント駆動型プログラムでも 3 つの分類を適用可能である。

このような分類やプログラム要件の粒度は、“ソフトウェア開発に最適な解法が 1 つしかないことはない[12]”といわれるように教材を作成する教授者の観点によって異なると考えられる。仮に相違があったとしても、異なる観点の中でプログラム要件とソースコードの関係を理解することは、初学者の柔軟な思考を育成するうえで重要であると考えられる。したがって、この相違は大きな問題にならないと考えられる。

4.2 実験結果

(1) レポート提出率と平均評点

図 6 に全 12 回分の課題出題に対するレポート提出率と平均評点を示す。試作システムを導入した講義において、実際に試作システムを利用した受講者は 30.5% (第 11 回が 36.6%, 第 12 回が 24.4%) と少なかった。利用しなかった学生には、試作システムにログオンせず、しかもレポート提出をしていない者が 54.5% もおり、レポート提出率の改善には至らなかった。

しかし、課題内容が難しくなっているにもかかわらず平均評点は向上している。試作システム導入の効果が現れたものと考えられる。

【課題内容】
赤・緑・青の 3 つのチェックボックスを選択すると、その色の円を描く Java アプレットを作成しなさい。

【プログラム要件整理の問題】
次のうち本プログラムの要件として適切なものを選び。
・ ボタンが押されたら色を代えて円を描画する
・ 3 つのチェックボックスの ON/OFF にしたがって、描画する図形を変更する
・ 3 つのチェックボックスの状態にしたがって、3 色の円を描画する
・ 3 つのチェックボックスの状態にしたがって、8 色の円を描画する → 正解選択肢

【要件タイプの選択問題】
次の要件を“入力処理”“計算などの処理”“出力処理”に分類せよ。
・ 「チェックボックス配置」 → 正解：入力処理
色指定のため 3 つのチェックボックスを配置する処理
・ 「チェックボックスイベント」 → 正解：入力処理
チェックボックスで発生したイベントを処理
・ 「円を描画」 → 正解：出力処理
アプレット画面上に円を描画する処理
・ 「描画色管理」 → 正解：計算などの処理
チェックボックスで指定された設定に従って円の色を管理する処理
・ 「色の決定」 → 正解：計算などの処理
チェックボックスの ON/OFF にしたがって描画色の値を決定する処理

【要件ごとの問題ページの問題例】
円を描画するために定義すべきメソッドは何か？
・ public void init()
・ public void itemStateChanged(ItemEvent e)
・ public void paint(Graphics g) → 正解選択肢
・ public void paint()

図 5 各問題ページの例

(2) 要件分類別の理解状況の違い

表 1 に問題ページ種別ごとの正答率の平均と閲覧状況（問題表示回数と採点回数）を示す。正答率は、学習者の理解状況を把握するために、各問題ページの最初の解答を対象にして算出している。

要件の整理及び要件タイプの選択問題は、他の問題ページに比べて正答率が低く、学習支援が必要であると考えられる。入力処理要件、計算などの処理要件、出力処理要件の各問題ページでは、正答率の差はほとんど生じなかった。しかし、出力処理要件の表示回数が少ないことから、出力処理要件は学習支援のニーズが比較的少ないと判断できる。入力処理要件及び計算などの処理要件に学習支援が必要であると考えられる。

(3) ページ網羅率とレポート評価

表 2 に、試作システムを利用した学生ごとのページ網羅率と該当講義のレポート評価を示す。ページ網羅率とは、

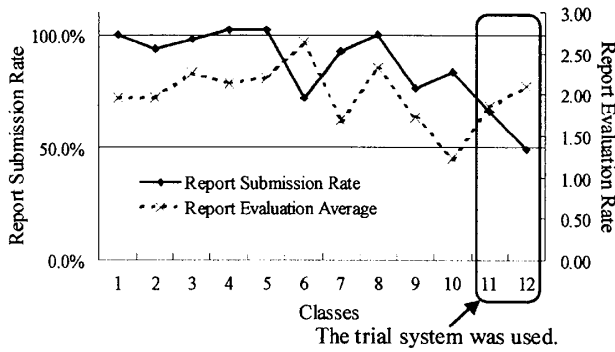


図6 レポート提出率と平均評点

あるプログラム要件に関する問題ページにアクセスしたとき、その要件に関する全ての問題ページに回答した場合に1となるように算出したものである。

各学生のページ網羅率の平均値とレポート評価の相関を算出したところ0.29となった。すなわち、ページ網羅率が高いほどレポート評価が向上する傾向がある。

各学生のページ網羅率の標準偏差とレポート評価では-0.50となり比較的強い負の相関となった。つまり、ページ網羅率の標準偏差が高いほどレポート評価が低い傾向がある。偏差値が高いということは、3つの処理分類間でページ網羅率の偏りが強いことを示している。このことから、学習の偏りがある学生に対して学習すべき箇所を案内するなどの支援が必要である。

5. むすび

本論文では、プログラミング初学者を対象とした学習モデルとして“RPRaS”を提案した。RPRaSは、学習の順序や支援のタイミングが構造化プログラミング手法に基づいている。また、プログラムの要件を3つに分類することによって、プログラム要件とプログラムソースの関連を容易に理解できるようにしている。

RPRaSに基づく学習支援方法を検討するため、本論文では基礎研究結果についても報告した。基礎研究のために試作システムをWebアプリケーションとして構築した。試作システムを用いた実験により、学習支援が必要と思われる箇所について検討した。

今後も試作システムを用いた評価を行っていく予定である。得られた評価結果から学習支援方法を検討し、本モデルによる学習効果をさらに向上させたいと考えている。

謝辞 本研究の一部は科学研究費補助金(研究課題番号:19500823)の助成を受けて実施した。

参考文献

[1] Arai,M., Yamazaki,T., Design of a Learning Support System to Aid Novice Programmers in Obtaining the Capability of Tracing. Proceedings of the Sixth IEEE International Conference on Advanced Learning Technologies, pp.396-398, 2006.

[2] Terada,M., ETV: a program trace player for students. Proceedings of the 10th annual SIGCSE conference on Innovation and technology in computer science education, 118 - 122, 2005.

[3] Iain,M., and Glenn.R., OGRE: Three-Dimensional Program Visualization for Novice Programmers. *Education*

表1 各種問題ページの正答率と閲覧状況の平均

問題ページ種別	正答率	表示回数	採点回数
要件の整理	0.49	65.00	42.67
要件タイプの選択	0.46	52.00	39.67
入力処理要件	0.69	43.33	19.78
計算などの処理要件	0.69	28.91	16.82
出力処理要件	0.64	24.71	15.88

表2 ページ網羅率とレポート評価

学生	ページ網羅率					レポート評価(*)
	入力処理	計算などの処理	出力処理	平均	標準偏差	
1	0.13	0.10	0.13	0.12	0.02	1.00
2	0.04	0.00	0.00	0.01	0.02	2.00
3	0.29	0.38	0.28	0.32	0.06	2.00
4	0.38	0.23	0.00	0.20	0.19	0.00
5	0.75	0.75	0.67	0.72	0.05	2.50
6	0.38	0.29	0.13	0.27	0.13	0.00
7	0.82	0.38	0.38	0.53	0.25	0.00
8	0.38	0.38	0.38	0.38	0.00	2.00
9	0.07	0.06	0.13	0.09	0.04	1.00
10	0.38	0.38	0.38	0.38	0.00	2.00
11	0.88	0.88	0.79	0.85	0.05	2.00
12	0.02	0.00	0.00	0.01	0.01	0.00
13	0.14	0.13	0.21	0.16	0.04	2.00
14	0.38	0.38	0.24	0.33	0.08	2.50
15	0.07	0.04	0.06	0.06	0.02	2.00
16	0.00	0.13	0.00	0.04	0.08	2.00
17	0.88	0.88	0.88	0.88	0.00	2.00
18	0.13	0.13	0.06	0.11	0.04	1.00
19	0.34	0.29	0.25	0.29	0.05	2.50
20	0.13	0.13	0.13	0.13	0.00	2.00
21	0.07	0.10	0.13	0.10	0.03	0.00

(*) 3点満点

and Information Technologies, Vol.9, No.3, pp.219-237, 2004.

[4] Matthew,C.,Jadud, A First Look at Novice Compilation Behaviour Using BlueJ. *Computer Science Education*, Vol.15, No.1, pp.25-40, 2005.

[5] Vassilios,Efopoulos, Vassilios,Dagdilelis, Georgios,Evangalidis, Maya Satratzemi, WIPE: a programming environment for novices. Proceedings of the 10th annual SIGCSE conference on Innovation and technology in computer science education, pp.113-117, 2005.

[6] Yasuhiko,Morimoto, Kunimi,Kurasawa, Setsuo,Yokoyama, Maomi,Ueno, and Youzou,Miyadera, A Support System for Teaching Computer Programming Based on the Analysis of Compilation Errors. Proceedings of the Sixth IEEE International Conference on Advanced Learning Technologies, pp.103-105, 2006.

[7] Chiken,K., Hazeyama,A., and Miyadera,Y., A programming Learning Environment Focusing on Failure Knowledge. *The IEICE Transactions on Information and Systems (Japanese edition)*, Vol.J88-DI, No.1, pp66-75, 2005.

[8] Matthew,C.,Jadud, Methods and tools for exploring novice compilation behaviour.Proceedings of the 2006 international workshop on Computing education research, pp.73-84, 2006.

[9] H.,D.,Mills, R.,C.,Linger and A.,R.,Herner, Box structured information systems. *IBM Systems Journal*, Vol.26, No.4, pp.395-413, 1987.

[10] Alan,R.,Hevner and Harlan,D.,Mills, Box-structured methods for systems development with objects. *IBM Systems Journal*, Vol.32, No.2, pp.232-251, 1993.

[11] Tom Demarco, Timothy Lister, *Software State of the Art: Selected Papers*. Dorset House, 1990.

[12] Robert L. Glass, *Facts and Fallacies of software Engineering*, Pearson Education Inc., 2003.