

協調プログラミング学習支援のための
分散および対面協調学習の分析

平井 佑樹

図書館情報メディア研究科

筑波大学

2012年4月

**Analysis of Collaborative Learning to Support
Collaborative Programming Learning**

Yuki Hirai

**Graduate School of Library,
Information and Media Studies
University of Tsukuba**

April, 2012

概要

プログラム言語やその文法等を学習するプログラミング学習がある。また、プログラミング学習において、学習者同士で協動的にプログラミング学習をする方法がある。この学習では、2名以上の学習者グループ内で、意見交換や相互に知識を補足しあう活動等が行われる。学習を支援することを学習支援といい、学習支援を目的とした、コンピュータを用いた環境やシステムのことを学習支援環境または学習支援システムと呼ぶ。学習支援環境を開発するためには学習者の学習形態、学習状態、学習目的等を考慮する必要がある。支援環境を開発するためには、それとは別に、既存の支援環境の利用において得られた知見を調査・分析する必要がある。

本研究では、協動的にプログラミング学習をする「協調プログラミング学習」を支援することを考え、協調プログラミング学習支援のために、既存の支援環境の利用において得られた知見の調査・分析を行った。本研究では、特に、分散環境での支援と対面環境での支援という観点で分析を行った。

既存の支援環境に関する調査を行った結果、問題点を2点挙げた。1点目は、分散環境における支援例が少ないことである。分散環境で大規模の人数で学習を進めていくような教育システムやeラーニングシステムが今後開発されるといわれているが、協調プログラミング学習に関して言えば、その実践例は多くない。分散環境における学習も今後普及していく可能性があることを考慮すると、支援環境開発のためには、分散環境で大規模の人数で行われる学習についても考慮する必要がある。2点目はプログラミング中の学習がうまくいかなかった学習者（学習不振者）に関する考察の少なさである。既存の支援環境を扱った研究では、個人で学習するよりも「エラーが少なくなった」、「プログラムの質が上がった」など、協調プログラミング学習を支持する結論を述べているが、学習不振者に関する考察は十分行われていない。支援環境開発のためには、学習不振者についても考慮していく必要がある。

本研究で挙げた問題を解決するため、本研究では、分散環境における協調プログラミング学習と対面環境における協調プログラミング学習を実施し、それぞれの学習において、学習者同士のやり取りや学習者の成績について調査・分析した。

分散環境における協調プログラミング学習として、分散非同期環境で利用する作問学習支援システムにおける学習を分析した。作問学習支援システムとは、問題を作成することによって学習をすることを対象として支援するシステムである。このシステムでは学習者がプログラミング学習に関連する問題（クイズ）を作成して、問題をシステムに蓄積し、

他学習者がシステムに蓄積された問題に解答することができる。問題に解答した学習者は、解答した問題について、例えば、問題文やその問題の答えに間違いがないという問題の正確性等を評価することができる。分散非同期環境で利用する作問学習支援システムにおける学習の分析では、問題の作成数の多さ、あるいは作成した問題の正確性の高さやテストの成績とに関係があったことを明らかにした。

対面環境における協調プログラミング学習として、対面同期環境で実施するペアプログラミングにおける学習を分析した。ペアプログラミングとは、あるプログラム作成課題について、2人1組で1台のコンピュータを共有してその課題に取り組む活動である。その活動を利用した学習をペアプログラミング学習という。ペアプログラミングでは、2人で相談しながら課題に取り組むことができるので、分析では特に課題解決中のペア同士の会話に注目した。この学習の分析では、ペアプログラミング中に発生するつまづきの解決に失敗した事例について、成功した事例と比較したときに、1回の発話長が長かったこと、説明を繰り返す回数が多かったことを明らかにした。

本研究で明らかにしたことを利用することで、学習に効果的な支援環境を開発することができる可能性がある。分散環境における協調プログラミング学習の分析で得られた知見を利用して分散環境における学習支援環境を開発するというに限らず、例えば、分散環境における学習支援環境に対面環境で得られた知見を加味したり、対面環境における学習支援環境に分散環境で得られた知見を加味したりすることで、学習に効果的な支援環境を開発することができる可能性がある。

現在も、計算機や情報ネットワーク技術は急速に進展を続けており、小中学校や高校、大学などの学校教育でも授業時間内の対面環境における学習だけでなく、eラーニングを利用した授業時間外の分散環境における学習も積極的に行われている。本研究では、分散環境での支援と対面環境での支援についてまとめ、それぞれの特徴を明らかにした。本研究で明らかにしたことは、協調プログラミング学習支援を考案する上では非常に参考となる結果である。協調プログラミング学習支援に限らず、例えば、協調学習支援、プログラミング学習支援等を考案する上でも参考となるだろう。

Abstract

The ability to understand the grammar of a programming language, to write a program, and to assemble an algorithm, is required in programming education. When a learner actually creates a program, some problems typically occur, even if the grammar and a (relatively easy) example of the program language are understood. In programming education, numerous practices, including the support of problem-solving, have been developed to date. Education and study methods have also received considerable attention.

"Collaborative Programming Learning Support" is focused on in this study. A learner learns programming language with his/her colleagues in collaborative programming learning. In this study, to support the collaborative programming learning, the author investigated how learners used a learning support system, how learners learned in the learning support system, or how the learning effectiveness was revealed in the application of learning support system. To analyze them, two collaborative programming learning of "Question-posing learning" and "Pair-programming learning" was conducted.

Question-posing is higher cognitive activity than question-answering and has positive effect on the learners' problem solving skills. Question-posing learning activity includes question-posing, question-answering, question-assessing and some discussion. These activities improve the learners' comprehension. In this study, the learners' activities in Concerto III were analyzed and the learning effectiveness in using the system was identified. The effectiveness of activities of posing questions has been discussed. The positive effectiveness of making high-quality question-items have been reported. The question-items consist of question-stem, the answer and the explanation. The high-quality question-items define as "the question-item where there are few mistakes, such as wrong answer and misspelling".

The programming method called 'pair-programming' originated in industry as a key component of the Extreme Programming (XP) development methodology. As the name suggests, two programmers work together at the same machine while developing code. One programmer (the driver) operates the keyboard and focuses on entering code, while the other programmer (the navigator) observes the work of the driver and offers suggestions in the code. The programmers regularly exchange roles. In this study, pair-programming was conducted in an introductory programming course. Success and failure cases in pair-programming were compared. In the comparison, the author focused on the conversation between the pair in pair-programming. In the failure cases, it was found that speech

length was long, and there was a great deal of repeated explanation and/or continuous speech. These insights indicate ways to determine whether collaborative work in pair-programming is progressing smoothly, and to assess the quality of collaborative work.

This research objective in broader sense is to support programming learning. Pair programming has been focused as one of the promising techniques of programming learning. The author does not intend to just using pair programming. The author intends to expand pair programming to computer-supported pair programming (CSPP). This means that a computerized environment (not the computer used for programming basically) senses the learning status of the pair, and once the environment senses something wrong with the pair it intervenes in the learning. This could be a mixture of ICAI (Intelligent Computer-Aided Instruction) and CSCL (Computer Supported Collaborative Learning) under the ubiquitous computing technology. To realize such CSPP, The author thoughts the author needs some symptoms to indicate the status of programming learning. This led to the study in this paper. So based on the findings presented in this paper, a CSPP system that use the symptoms will be able to develop.

目次

第1章 序論.....	1
1.1 研究背景.....	1
1.1.1 学習.....	1
1.1.2 学習支援.....	1
1.1.3 協調学習支援.....	3
1.1.4 プログラミング学習支援.....	4
1.1.5 協調プログラミング学習支援.....	5
1.2 研究目的.....	7
1.3 研究意義.....	7
1.4 研究成果.....	8
1.5 論文構成.....	9
第2章 関連研究.....	10
2.1 既存研究の整理.....	10
2.1.1 支援対象の拡大とそれに応じた支援.....	10
2.1.2 協調学習支援.....	12
2.1.3 プログラミング学習支援.....	14
2.1.4 協調プログラミング学習支援.....	16
2.2 本研究の位置づけ.....	19
2.2.1 まとめ方針.....	19
2.2.2 既存研究のまとめ.....	20
2.2.3 既存研究の問題点.....	21
2.2.4 問題点の解決.....	22
第3章 分散環境における協調プログラミング学習の分析.....	24
3.1 研究概要.....	24
3.2 関連研究.....	26
3.2.1 システムの機能・運用・評価.....	26
3.2.2 作問学習支援システムにおける学習者の活動分析.....	28
3.2.3 データマイニング手法.....	29
3.3 作問学習支援システム.....	29
3.3.1 システムの利用プロセス.....	29

3.3.2	Concerto III の概要.....	30
3.4	システム適用実験	34
3.4.1	実験概要	34
3.4.2	実験結果	34
3.4.3	システム利用実験から得られるデータ	38
3.5	問題作成数および作成した問題の正確性と成績との関連	38
3.5.1	問題作成数を用いた試験結果の分析.....	38
3.5.2	問題の正確性評価データを用いた試験結果の分析.....	40
3.5.3	システム利用アンケート.....	43
3.6	問題作成数, 問題の正確性を用いた分析に対する考察.....	44
3.7	相関ルールマイニングを利用した分析.....	45
3.7.1	相関ルールの抽出.....	46
3.7.2	抽出結果	48
3.8	相関ルールマイニングを用いた分析に対する考察	49
3.8.1	既存研究との比較.....	49
3.8.2	相関ルールマイニングを行うことの利点	50
3.8.3	相関ルールマイニングの限界.....	51
3.8.4	作問学習支援システムを用いることの効果.....	51
3.9	まとめ.....	53
第4章	対面環境における協調プログラミング学習の分析.....	54
4.1	研究概要	54
4.2	関連研究	55
4.2.1	プログラミング導入教育におけるペアプログラミング	55
4.2.2	ペアプログラミングにおけるペア同士のコミュニケーション分析	55
4.2.3	ペアプログラミングにおける会話の役割	56
4.3	ペアプログラミング演習	56
4.3.1	演習設定	56
4.3.2	データ収集.....	59
4.4	発話分析	61
4.4.1	データ処理.....	61
4.4.2	分析方法	62
4.5	分析結果	66
4.5.1	発話量.....	66
4.5.2	説明の繰り返し	67
4.5.3	ペアの両者が交互に発話できているか.....	68
4.6	考察	68

4.6.1	作業中のペア同士のやり取り	68
4.6.2	本研究の限界	69
4.6.3	協調作業の状態推定	70
4.7	まとめ	70
第5章	総合検討	72
5.1	本研究で得られた知見のまとめ	72
5.2	検討	73
5.2.1	分散環境における協調プログラミング学習支援	73
5.2.2	対面環境における協調プログラミング学習支援	77
5.2.3	今後の課題	80
5.3	将来の展望	81
第6章	結論	85
	謝辞	87
	参考文献	88
	研究業績	97

図目次

図 1-1	ITS の 1 つであるモンサクンで表示される画面 [平嶋, 2005a]	2
図 1-2	学習者が解答した内容に誤りがあった場合にそれを指摘する例 [平嶋, 2005a]	3
図 1-3	概念的 CSCL 環境 [稲葉, 1999]	4
図 1-4	プログラミング学習支援システムの例 [伊藤, 2000]	5
図 1-5	グループ学習課題の例 [生田目, 2004]	6
図 3-1	システムの利用プロセス	30
図 3-2	問題作成インタフェース	31
図 3-3	問題評価インタフェース	32
図 3-4	作問要求インタフェース	33
図 3-5	プレテストとポストテストの成績分布	35
図 3-6	ConcertoIII の問題作成数の多少による中間テスト成績の分布	40
図 3-7	ConcertoIII の問題作成数の多少による期末テスト成績の分布	40
図 3-8	学習者による問題の正確性評価と教授者による問題の正確性評価の散布図	41
図 3-9	ConcertoIII の問題の正確性評価平均の高低による中間テスト成績の分布	42
図 3-10	ConcertoIII の問題の正確性評価平均の高低による期末テスト成績の分布	42
図 4-1	ペアプログラミング演習で用いた課題の例	58
図 4-2	データ取得用カメラの設置位置	58
図 4-3	ペアプログラミング演習で撮影した映像データの一場面	58
図 4-4-1	Driver と Navigator が完全に交互に発話している事例	63
図 4-4-2	Driver と Navigator が完全に交互に発話していない事例	64
図 4-4-3	Driver と Navigator が発話交替しているところと発話交替していないところがある事例	64
図 4-5	1 人あたりの発話時間	66
図 4-6	1 人あたりの発話頻度	67
図 4-7	1 人あたりの発話長	67
図 4-8	説明を繰り返した回数 (発話数に対する比率)	68
図 4-9	連続発話率	68
図 5-1	対面同期環境における協調プログラミング学習支援の構想	82
図 5-2	分散同期環境における協調プログラミング学習支援の構想	83

表目次

表 2-1	グループウェアの分類 [農林水産省, 1996]	19
表 2-2	本研究におけるグループウェアの分類.....	20
表 2-3	既存研究のまとめ	21
表 2-4	本研究で実践した学習の分類.....	23
表 3-1	本研究と関連研究の比較	27
表 3-2	ConcertoⅢの利用状況.....	35
表 3-3	問題とその問題に関する議論の例.....	36
表 3-4	ConcertoⅢの問題作成数の多少による中間テスト結果の分析.....	39
表 3-5	ConcertoⅢの問題作成数の多少による期末テスト結果の分析.....	40
表 3-6	ConcertoⅢの問題の正確性評価平均の高低による中間テスト結果の分析	42
表 3-7	ConcertoⅢの問題の正確性評価平均の高低による期末テスト結果の分析	42
表 3-8	行動履歴の集計	47
表 3-9	集計結果の 2 値化	47
表 3-10	成績の向上と関連するルール	48
表 3-11	成績の低下と関連するルール	49
表 4-1	ペアプログラミング演習中に発生したつまずきの事例	60
表 4-2	ペアプログラミング演習で収集したコミュニケーションデータ	61
表 4-3	失敗事例 B で行われた会話の一部	65

第1章 序論

1.1 研究背景

1.1.1 学習

日常生活において「学習」という言葉をよく耳にする。日常、学習という言葉は、“子供たちが勉強して何かを覚える”というようなニュアンスで使っていることが多くあるが、本研究でいう「学習(learning)」は日常的な意味とは少し違う。教育や学習に関わる用語を解説している教育工学事典[教育工学, 2000]によれば、学習とは“個人ないしは個人を含む集団が自ら行為をなすことによって、さまざまな知識を獲得する過程”としている。「勉強して何かを覚える」ということはもちろん学習であるが、何かを覚えるまでの過程（プロセス）も学習に含まれる。

現実世界において、学習者は教科書などを通して独自に知識を獲得する個別学習を中心に、学習状態や学習目的に合わせて他の学習形態に推移することで効果的に知識を獲得する[小尻, 2003]。学習形態には、例えば、数人程度を単位とした小集団学習の形態を指すグループ学習[教育工学, 2000]や、大学の講義等で行われる一斉学習などがある。

効果的に知識を獲得する方法は学習形態に限らず、学習する方法や手段によることもある。学習する方法では、例えば、観察による学習がある。観察による学習とは、自らが直接に行わず、他者の行為を観察するだけで、当該の知識を獲得することである[教育工学, 2000]。学習者は、個別学習のような自分で問題を解決する体験を通じてのみならず、他者の問題解決行動や振る舞いの観察を通じて間接的にも知識やスキルを獲得しうることもある[Bandura, 1971]。学習する手段では、例えば、eラーニングがある。eラーニングの“e”はelectronicの意味であり、コンピュータ等の情報技術を用いて行う学習をすることである。計算機や情報ネットワークは現在も急速に進展を続けており、小中学校や高校、大学などの学校教育でもeラーニングが行われている。

1.1.2 学習支援

前項の学習を支援することを学習支援という。学習支援を目的とした、コンピュータを用いた環境やシステムのことを学習支援システムと呼び[教育工学, 2000]、学習状態や学習目的に合わせて様々なシステムが開発されている。

学習支援システムには、LMS(Learning Management System)や WBT (Web-Based Training)と呼ばれるインターネットを利用して家庭や職場などで学習を行うことができるようなシステムがある。これらのシステムを利用することにより、小中学校や高校、大学等での授業や講義外の時間でも、インターネットを利用して学習することができる。

学習支援システムの別の例として、ITS (Intelligent Tutoring System) と呼ばれる学習者の個性に応じて教授をするようなシステムがある。図 1-1 は ITS の一つであるモンサクン[平嶋, 2005a]で表示される画面である。このシステムでは学習者が問題を作ることによって学習する(この学習については3章で触れる)ことを支援するシステムである。図 1-1 の左側には、上部に作成する問題の目標となる2項演算が示され、その下に単文カードを入れることができる空欄が3つ用意されている。右側には、文章題を完成させるための単文カードが用意されており、これらのカードはドラッグ&ドロップによって画面上を自由に動かすことが可能となっている。左側下部には答え合わせボタンが用意されており、文章題を完成させると、ボタンを押すことができる。図 1-2 は答え合わせボタンの後に表示される画面の例である。モンサクンでは、答え合わせボタンが押された後に、作成された文章題を診断し、問題が適切であった場合はその旨を伝え、適切でなかった場合は、診断結果に基づいて修正を誘導するコメントを提示する。図 1-2 では画面左部で作成された文章題が作成する問題の目標となる式にあわないため、作成した文章題が間違いであることを学習者に知らせている。ITSのように学習支援システムを用いた個別学習環境に学習指導を実現する教師的役割を導入することで、システムと学習者のやり取りが実現され、学習支援システムを用いた学習効果が著しく向上している[小尻, 2003]。

学習支援システムは、学習者の学習状態や学習目的に合わせて開発する必要がある。情報技術の発達により、学習者が同じ場所の学習環境に存在するような対面環境のみではなく、距離的に離れた場所に存在する学習者同士でやり取りするような分散環境での学習も可能となっている今日では、学習状態や学習目的だけでなく、今後増え続けるであろう学習形態も考慮しつつ、学習支援システムを開発する必要がある。

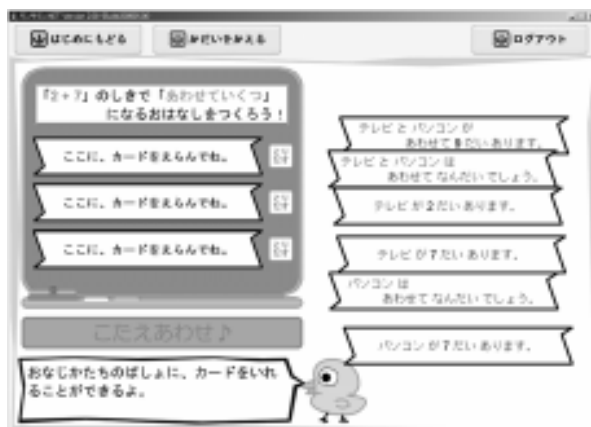


図 1-1 ITS の一つであるモンサクンで表示される画面 [平嶋, 2005a]

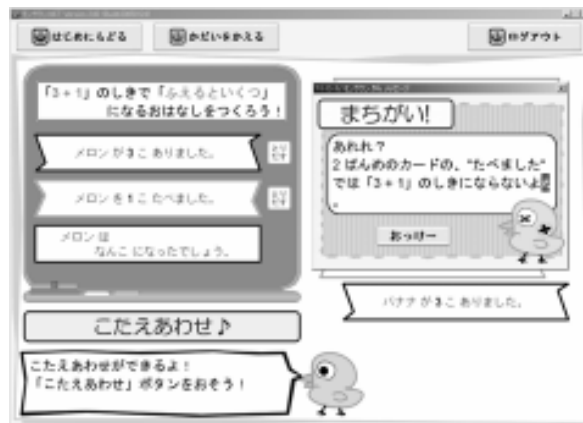


図 1-2 学習者が解答した内容に誤りがあった場合にそれを指摘する例 [平嶋, 2005a]

1.1.3 協調学習支援

学習の 1 つに、協調学習と呼ばれる学習がある。これはグループ学習の一種であるが、特に、同一の課題について参加者が意見交換、競合、交渉、合意形成等を繰り返し、グループの合意としての成果を出すことを協調学習という[稲葉, 1999]。仲間とともに問題解決や学習を行うという状況は、我々にとって日常的であり自然である。仲間同士はお互いに教えあったり、競い合ったりして学習を進めていく。そこでは、対象となる問題が解決されるだけでなく、自己表現力の育成、他者と自分との意見の相違点を整理するような能力の育成も期待される。仲間と協調的に学習することによる豊富な学習効果は教育分野においては古くから認識されてきている[稲葉, 1999]。

協調学習を支援するシステムを協調学習支援システムと呼び、CSCL (Computer-Supported Collaborative Learning) と呼ばれる研究領域で、近年盛んに研究が行われている。図 1-3 は CSCL 環境を概念的に示したものである[稲葉, 1999]。環境は参加者と使用するツールとで構成される。参加者には、複数の学習者に加えて、指導者的な役割が含まれる場合もある。ある学習者に着目した場合に、他の学習者は協調学習者と呼ばれる。学習者は、ネットワークを介して協調学習者とともに学習や問題解決を行う。ここでの指導者は、従来の「正しい知識の保持者」としての役割よりもむしろ参加者間の相互作用のコーディネーションに重点がおかれる[稲葉, 1999]。

協調学習支援システムは学習支援システムの種類であるため、学習形態、学習状態、学習目的に合わせて適切にシステムを開発する必要がある。それに加え、学習者と協調学習者のやり取りをスムーズにできるようにする等、参加者間のコミュニケーションについても考慮する必要がある。

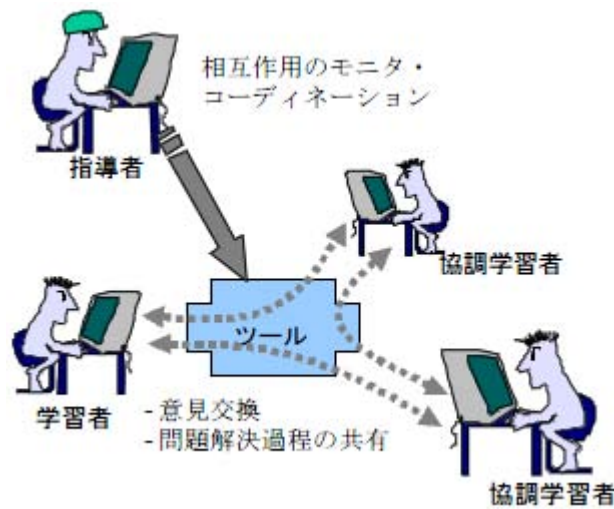


図 1-3 概念的 CSCL 環境 [稲葉, 1999]

1.1.4 プログラミング学習支援

プログラミングはコンピュータ等のプログラムを作成することを指し、プログラムはコンピュータに対してなんらかの「仕事」を指示するためのものを指す[高橋, 2007]。プログラミングについて学習することをプログラミング学習と呼ぶ。プログラミング学習では、コンピュータに指示を与えるためのプログラムを作成するために、指示を与えるための言語やその言語の文法を学習する。それに加え、例えば、アルゴリズムと呼ばれるコンピュータに指示を与える方法や手順について学習することや、アルゴリズムを視覚的に表現するためのフローチャート（流れ図）について学習することもある。

プログラミング学習では、プログラム言語の文法やプログラム書法を理解する能力とアルゴリズムを組み立てる能力が要求される。その能力を育成するためのプログラミング学習では、学習者が計画性を持って自律的にプログラミングをしてくことで効果があがると言われている[匂坂, 2009]。

プログラミング学習を支援することをプログラミング学習支援と呼び、そのための環境をプログラミング学習支援システムと呼ぶ。プログラミング学習支援のためのシステムはこれまでも多く提案・利用されている。例えば、プログラムを作成するための支援をするデバッガ（プログラムの不具合の発見やその修正を補助するもの）やトレーサ（プログラムの実行過程を示して、プログラムの修正を補助するもの）がある。また、アルゴリズムをアニメーションさせることでプログラムの流れを分かりやすくするシステムや、プログラムの不具合を診断し、その結果に応じて学習者の指導を行うという ITS もある。

図 1-4 はあるプログラミング学習支援システムで表示される画面である[伊藤, 2000]。画面右側に学習者が入力したプログラムが表示され、画面左側には入力したプログラムに対する説明が表示される。この説明はシステムが、学習者が入力したプログラムを解析して、

システムが自動で生成している。画面左側の説明では、はじめ、「プログラムの説明を始める」と書かれたボタンのみ示されており、学習者がこのボタンをクリックすると図 1-4 中の 1 と 2 の説明が現れる。これらの説明ボタンをクリックされると、2-1、2-2、2-3 のような動作の詳細説明が提示される。このように提示することで、学習者はアルゴリズムのどの部分がどの命令で実現されているかを見ることができる。

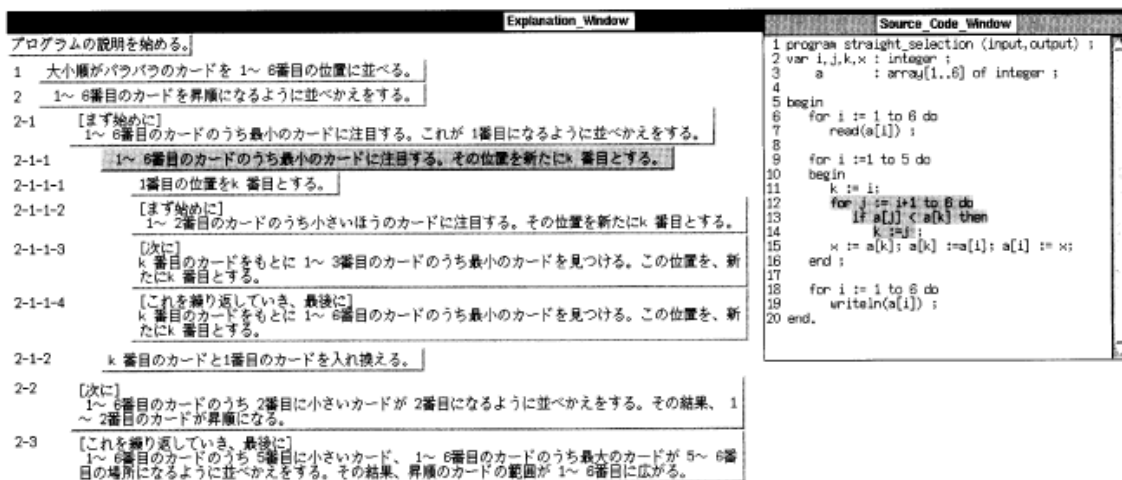


図 1-4 プログラミング学習支援システムの例 [伊藤, 2000]

現実世界におけるプログラミング学習支援の 1 つの目標は、学習者が自律的にプログラミングやプログラミング学習させることである。学習者が自律的にプログラミングをしていくことは重要ではあるが、中には、プログラム言語の文法や簡単な例題を理解することができても、実際にプログラムを作成するとなったとき、プログラムを組むことができない[新開, 2009]等の問題が生じ、学業不振になってしまう学習者もいる。このような学習者を学習過程の早い段階で見つけ出し、弱点補強のための練習問題の提供や、学習方法に関するアドバイスの提供等、具体的な支援を行うことが望まれている[匂坂, 2009]。

1.1.5 協調プログラミング学習支援

プログラミング学習において協調学習を行うことを、協調プログラミング学習と呼ぶ。つまり、協調プログラミング学習とは、プログラミング学習について参加者が意見交換、競争、交渉、合意形成等を繰り返し、グループの合意としての成果を出すことである。この学習をすることで、例えば、グループ内の学習者間で、理解度や進度、学習意欲に差があるような状況では、各自の足りない部分について、お互いに教えあい不足した知識を補強し、獲得していくことができる。また、自らが十分理解している知識については、他の学習者に解説することで、より深い理解を得られる[生田目, 2004]。

協調プログラミング学習を支援することを協調プログラミング学習支援と呼ぶ。これはプログラミング学習方法の1つである。協調プログラミング学習支援として、例えば、学習者同士でプログラムやフローチャートをピア・レビューする[生田目, 2004]というものがある。ここでは、グループ内の学習者同士で質問や相談しあいながら、課題のプログラムを制作する。そしてフローチャートとプログラムおよび実行結果については、ピア・レビューする。ピア・レビューは、プログラム設計者とレビュー（評価者）で行う。設計者がレビューに自分が製作したフローチャート（図1-5）等の設計物について解説する。

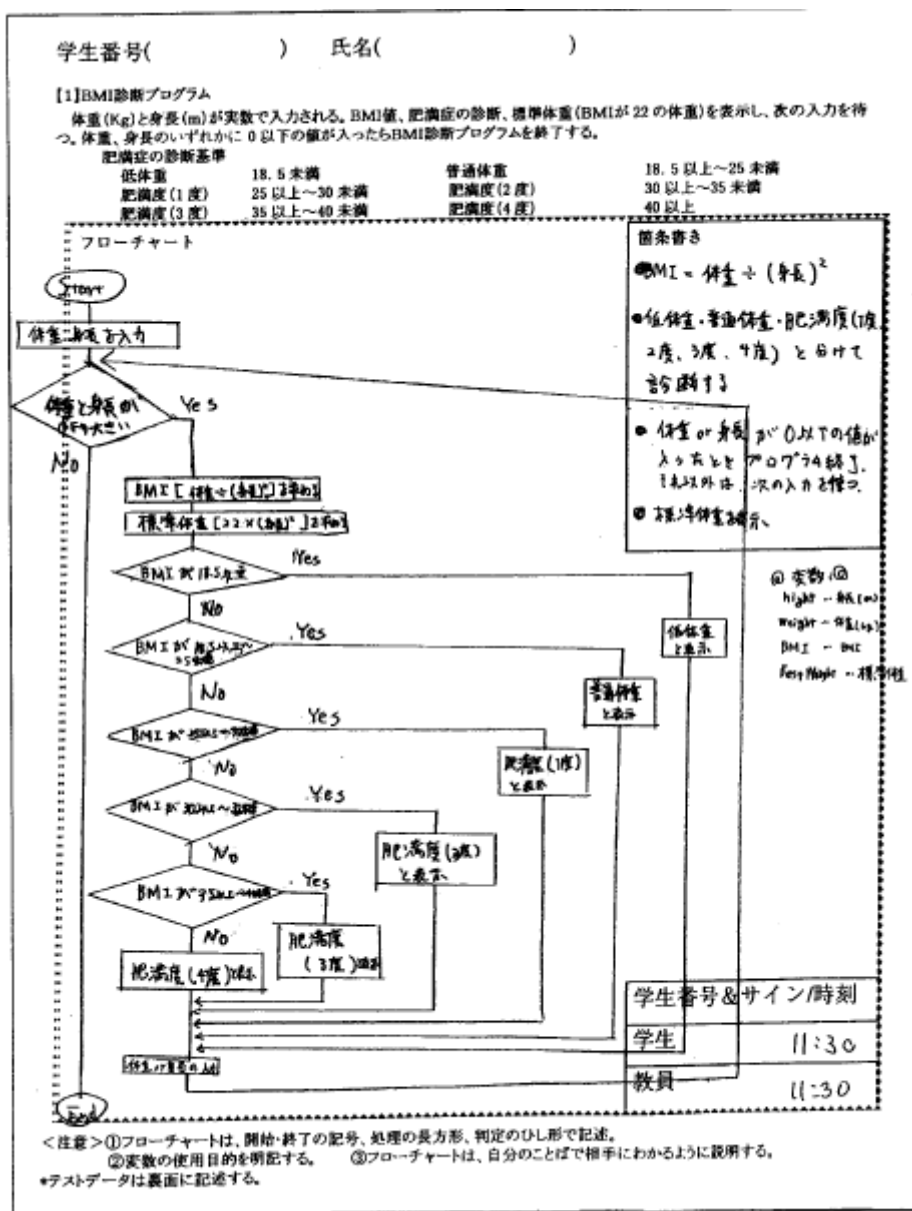


図 1-5 グループ学習課題の例 [生田目, 2004]

レビューは設計者に質疑をし、設計者はそれに答える。これにより、レビュー対象の設計物に誤り、抜け、冗長なものが含まれないことを確認する。

このように、協調プログラミング学習では、プログラミングに関する内容について、学習者同士で意見交換等を繰り返し行い、学習者グループとしての成果を出す。協調プログラミング学習支援では、プログラミング学習の側面のみではなく、協調学習の側面も考慮して支援する必要がある。

1.2 研究目的

研究背景でも述べたが、今日までの情報技術の発達により、提案される学習支援手法は多種多様化しており、今後も支援対象となる学習形態が拡大する可能性がある。これは協調学習支援、プログラミング学習支援、協調プログラミング学習支援にも言えることである。協調プログラミング学習支援について言えば、前節で述べたピア・レビューをプログラミング学習に導入する手法[生田目, 2004]や、C言語プログラミング教育を行うためにグループ活動・他者評価などを取り入れた授業展開を行う手法[新開, 2009]、プログラム学習に共同学習を取り入れることで、話し合い活動を取り入れる手法[山本, 2006]などがあり、現在も支援対象の拡大が進んでいる。学習支援環境を開発するためには学習者の学習形態、学習状態、学習目的等を考慮する必要がある。支援環境を開発するためには、それとは別に、既存の支援環境の利用において得られた知見を調査・分析する必要がある。

本研究では、協調的にプログラミング学習をする「協調プログラミング学習」を支援することを考え、協調プログラミング学習支援のために、既存の支援環境の利用において得られた知見の調査・分析を行った。本研究では、特に、分散環境での支援と対面環境での支援という観点で分析を行った。

1.3 研究意義

本研究の意義は分散環境での支援と対面環境での支援という観点で分析を行ったことである。現在も、計算機や情報ネットワークは現在も急速に進展を続けており、小中学校や高校、大学などの学校教育でも授業時間内の対面環境における学習だけでなく、eラーニングを利用した授業時間外の分散環境における学習も積極的に行われている。本研究では既存の分散環境における協調プログラミング学習、対面環境における協調プログラミング学習についてまとめた。その後、本研究では、分散環境および対面環境における協調プログラミング学習を実施し、文献調査等では明らかにならなかった点について明らかにした。これらを明らかにすることで、今後開発されるであろう協調プログラミング学習支援環境

やシステムにおいて分析結果が考慮され、分散環境、対面環境それぞれにおける学習支援を効果的に行うことができると期待される。

1.4 研究成果

本研究で既存の支援環境に関する調査を行った結果、問題点を2点挙げた。1点目は、分散環境における支援例が少ないことである。分散環境で大規模の人数で学習を進めていくような教育システムやeラーニングシステムが今後開発されるといわれているが、協調プログラミング学習に関して言えば、その実践例は多くない。分散環境における学習も今後普及していく可能性があることを考慮すると、支援環境開発のためには、分散環境で大規模の人数で行われる学習についても考慮する必要がある。2点目はプログラミング中の学習がうまくいかなかった学習者（学習不振者）に関する考察の少なさである。既存の支援環境を扱った研究では、個人で学習するよりも「エラーが少なくなった」、「プログラムの質が上がった」など、協調プログラミング学習を支持する結論を述べているが、学習不振者に関する考察は十分行われていない。支援環境開発のためには、学習不振者についても考慮していく必要がある。

以上の2つの問題を解決するため、本研究では、分散環境における協調プログラミング学習と対面環境における協調プログラミング学習を実施し、それぞれの学習において、学習者同士のやり取りや学習者の成績について調査・分析した。

分散環境における協調プログラミング学習として、分散非同期環境で利用する作問学習支援システムにおける学習を分析した[平井, 2010][Hirai, 2010]。作問学習支援システムとは、問題を作成することによって学習をする学習者を対象として支援するシステムである。このシステムでは学習者がプログラミング学習に関連する問題（クイズ）を作成して、問題をシステムに蓄積し、他学習者がシステムに蓄積された問題に解答することができる。問題に解答した学習者は、解答した問題について、例えば、問題文やその問題の答えに間違いがないという問題の正確性等を評価することができる。この学習の分析では、システム利用状況と学習者の試験成績に関連があるかどうかを調査した。その結果、問題の作成数の多さ、あるいは作成した問題の正確性の高さとテストの成績とに関係があったことを明らかにした。

対面環境における協調プログラミング学習として、対面同期環境で実施するペアプログラミングにおける学習を分析した[平井, 2012]。ペアプログラミングとは、あるプログラム作成課題について、2人1組で1台のコンピュータを共有してその課題に取り組む活動である。その活動を利用した学習をペアプログラミング学習という。ペアプログラミングでは、2人で相談しながら課題に取り組むことができるので、分析では特に課題解決中のペア同士の会話に注目した。この学習の分析では、ペアプログラミング中に発生するつまずきの解

決の成功・失敗がペアプログラミング中の会話と関連があるかどうかを調査した。その結果、失敗事例のほうが、成功した事例と比較したときに、1回の発話長が長かったこと、説明を繰り返す回数が多かったことを明らかにした。

本研究で明らかにしたことを利用することで、学習に効果的な支援環境を開発することができる可能性がある。分散環境における協調プログラミング学習の分析で得られた知見を利用して分散環境における学習支援環境を開発するというに限らず、例えば、分散環境における学習支援環境に対面環境で得られた知見を加味したり、対面環境における学習支援環境に分散環境で得られた知見を加味したりすることで、学習に効果的な支援環境を開発することができる可能性がある。

1.5 論文構成

第2章では、本研究の背景や目的・意義について、関連文献を引用しながら、詳しく説明する。その上で、本研究の位置づけを明らかにし、本研究の意義を再確認する。具体的には、協調プログラミング学習支援に関連する研究について述べ、本研究と関連研究の関係を明らかにし、その関係の中における本研究の意義について述べる。

第3章、第4章では、本研究で行った協調プログラミング学習の分析について述べる。第3章では、分散非同期環境で利用する作問学習支援システムにおけるプログラミング学習の分析について述べ、第4章では、対面同期環境で実施するペアプログラミング学習の分析について述べる。

第5章では、本研究で得られた結果をもとに、協調プログラミング学習支援について総合検討を行う。総合検討では、2章で述べた関連研究で得られた知見と本研究で得られた知見にどのような関係があるのか、本研究の知見が協調プログラミング学習支援のためにどのように役立つのかについて議論する。

最後に第6章では、研究の結論として本研究の貢献を整理し、研究全体をまとめる。

第2章 関連研究

本章では、序論で述べた本研究の背景や目的・意義について、関連文献を引用しながら、詳しく説明する。その上で、本研究の位置づけを明らかにし、本研究の意義を再確認する。具体的には、協調プログラミング学習支援に関連する研究について述べ、本研究と関連研究の関係を明らかにし、その関係の中における本研究の意義について述べる。

2.1 既存研究の整理

本節では、協調プログラミング学習支援に関連する研究について述べる。まず、支援対象が拡大していく可能性があることを述べる。その後、協調学習支援、プログラミング学習支援、それらを併せ持った協調プログラミング学習支援に関連する研究について述べる。

2.1.1 支援対象の拡大とそれに応じた支援

支援対象の拡大とそれに応じた支援については、小尻の論文[小尻, 2003]でよくまとめられている。次の4つの段落は、その論文から引用したものである。

教育支援システムは、計算機技術の発展に伴い、支援対象とする学習形態を様々に変化させてきた。1970年代からは、計算機の一般利用の増加に伴って、計算機を用いた個別学習を可能にするため、計算機内に学習コンテンツを構築することに焦点をあてた CAL (Computer Assisted Learning) が主に研究された。CAL では学習者がシステムの機能を理解してシステムを操作することで、自律的に知識を獲得する[Zusaku]ことができる。例えば、瀧口らは、Java などのオブジェクト指向プログラミングの基礎教育を行うため、オブジェクト指向の重要な概念を身につけさせるためのロールプレイング・ゲームを開発した[瀧口, 2002]。このシステムでは、ゲーム中で利用する魔法命令を作成する際に、継承やカプセル化などの概念を使用させることで、オブジェクト指向の概念を体験から学習させる。また、佐合らはゲームを楽しみながら自然に英会話能力を高めることを目的とした RPG 型英会話学習システムを構築している[佐合, 2000]。学習者はフィールド内に存在するキャラクターと英会話のドリル学習を行うことにより、問題を解決していく。このように、学習効果を狙ったシステムを計算機内に構築することで、計算機を個別学習のための媒体として定着させることに成功した。

1980年代になると、ローカルエリア・ネットワークの普及、また人工知能技術の発達も伴って、講義支援、すなわち集団教授を提供するための CAI (Computer Assisted Instruction)、さらには学習者の個性に応じた個別教授を実現するための知的教授システム、ITS (Intelligent Tutoring System) が注目されてきた[スリーマン, 1987]. CAI では教材内に教材データと教材提示の制御が含まれているのに対し、ITS では教材提示が教材とは独立した教授戦略部において制御される[溝口, 1995], [大槻 1993]. 教授戦略部では学習者の学習状態を表す学習者モデルを構築し、学習者モデルに応じて教材の提示方法を変更したり助言を生成することで、CAI に比べて柔軟な指導が可能となる. 仲林らは WWW (World Wide Web) を利用した個人適応型 CAI システム CALAT を構築している[仲林, 1997]. CALAT の WWW サーバはテキスト、アニメーション、音声から成るマルチメディア教材を保持しており、演習問題の解答内容に応じて次に選択する教材データに変化を持たせている. 一方、ITS を指向する研究に関して、Wing らは教授戦略を小さい教授単位に分割して再構成することで、一つの知的教授システム内で様々な教授戦略を実現する手法を提案している[Wing, 1998]. 様々な教授単位を組み合わせることにより、教授戦略作成者の負担を軽減しながら、多様な教授戦略を実現可能としている. また、堀口らは学習者モデルを基に学習者の理解状態を向上させるため、高校物理の力学の問題を対象に、学習者に誤りを気づかせるための問題空間のシミュレーション・システムを構築している[堀口, 2001]. このシステムは、学習者が立式した誤方程式を基に物体の不自然な動きをシミュレーションすることで、学習者に解法の誤りを認識させている. このように、学習支援システムを用いた個別学習環境に「学習指導」を実現する教師的役割を導入することで、システムと学習者との個別教授環境が実現され、学習支援システムを用いた学習効果が著しく向上した.

1990年代後半にはインターネットの急激な普及により、インターネットのワールドワイドな特徴を利用し、計算機内で他の学習者と共にグループで学習する集団学習を実現しようという試みがなされてきた. 集団学習の中で、特に共通の問題を保持する複数の学習者が議論を通じてお互いの知識を交換し、問題を解決することを目的とした学習形態は協調学習と呼ばれ、協調学習を支援するシステムは CSCL (Computer Supported Collaborative Learning) と呼ばれて盛んに研究されるようになった[先進学習基盤協議会, 2001]. Constantino-Gonzalez は ER (Entity-Relationship) 図の作成問題を対象とした協調学習環境において、個々の学習者にコーチエージェントを導入し、学習を支援する枠組みを明らかにした[Constantino-Gonzalez, 2000]. コーチエージェントは対応する学習者の作成した ER 図とグループの ER 図を比較し、相違点があればそれらを解消するように発言を促すことで、グループの議論を活性化させている. 一方、Heh らは team server, team manager, member の 3 階層で学習参加者を管理することで、多数の学習者が参加し、多数の学習グループが存在することのできる協調学習環境を構築している[Heh, 1996]. CSCL の発展に従って、教師からの一方的な知識の提供ではなく、学習者間の相互の意見交換による知識

の構築を計算機が支援するようになった。

教育支援システムは、最新の計算機技術を巧みに取り入れ、支援対象となる環境を拡大している。また、教育支援システムは、その発展に従って、現実世界の学習活動の補助的役割から、CSCLのように新しい学習環境を提案し、普及させるなど、現実世界の学習活動にその影響力を増大させつつある。情報ネットワークが普及し、学習データが全て計算機内に蓄積可能となった今日、学習者が同時に学習環境に存在するような同期環境のみではなく、ビデオなどを用いて時間的に異なった空間で情報を交換できるような非同期環境下での学習など、これまでにない学習形態での学習も想定される。従って、次世代の教育支援システムの構築には、最新の計算機技術を意識し、今後普及する学習形態を予測しながら、その学習効果を高めるような支援を提供する必要がある。

ここで述べられている教育支援システムは、本研究における学習支援システムと同義であると考えられる。小尻が述べたことをまとめると、次の4つになる。

- 1970年代では計算機の一般利用の増加に伴って、計算機を用いた個別学習をすることができるようにした。
- 1980年代ではローカルエリア・ネットワークの普及、また人工知能技術の発達も伴って、集団教授を提供するためのCAIや学習者の個性に応じた個別教授を実現するためのITSが注目された。
- 1990年代後半にはインターネットの急激な普及により、インターネットのワールドワイドな特徴を利用し、計算機内で他の学習者と共にグループで学習する集団学習を実現しようという試みがなされてきた。
- 次世代の教育支援システムの構築には、最新の計算機技術を意識し、今後普及する学習形態を予測しながら、その学習効果を高めるような支援を提供する必要がある。

本研究で扱う協調プログラミング学習は、プログラミング学習において協調学習を行うことであるため、集団学習の一部である。小尻の論文では、個別学習、個別教授、集団学習、集団教授のように、主に学習形態について触れている。情報技術の発達により、学習者が同じ場所の学習環境に存在するような対面環境のみではなく、距離的に離れた場所に存在する学習者同士でやり取りするような分散環境での学習も可能となっているように、新たな学習形態が普及しつつあるのは確かではある。学習支援環境を開発するためには、新たな学習形態を考慮することはもちろん必要である。それだけでなく、例えば、学習者の学習状態や学習目的に応じて支援の方法を変えることができるような柔軟な支援環境の開発が望まれる。

2.1.2 協調学習支援

協調学習支援とその意義については、稲葉らの論文[稲葉, 1999]でよくまとめられている。

これについて、稲葉は次のように述べている。

協調学習は、これはグループ学習の一種であるが、特に、同一の課題について参加者が意見交換、競合、交渉、合意形成等を繰り返し、グループの合意としての成果を出す。その過程では、個々の意見の調整、修正などが行われる。ここで重要視されるのは、最終的な成果を出すに至るまでのプロセスや、他者との相互作用自体の質である。最終的に問題が解決されたとしても、そこに至るまでのプロセスによって参加者が獲得する知識やスキルが異なるため、より効果的なプロセスを参加者に提供することを目的として支援環境が設計される。コンピュータ・ネットワークを介して学習を行うことにより、成果を上げるプロセスをすべての学習者が共有することが可能になった。さらにプロセスや学習者同士の相互作用を記録することも可能になり、それらの分析に基づく効果的な支援が期待される。協調学習支援では、それらを共有する環境のための利用しやすいインタフェースを提供することに加えて、環境において「何が」コミュニケーションされているのかに焦点をあて、人工知能技術を用いて積極的に参加者間の活動をコントロールしようとする試みも行われている。

また、稲葉は協調学習の有効性を示唆する理論的な背景や現象について、次の 8 点を挙げている。

- **Peer Tutoring:** Peer Tutoring は、学習者が相互に助け合い、教えることによって学習する教授体系である。Peer とは、当該領域において専門家でない集団に属するメンバーである。教えられる側の学習者には、知識の増加や、個別化された教授を得ることができるなどの利益の他に、仲間とともに学習することによる動機づけも期待される。また、教える側の学習者には、自己の知識を外化することによる知識の強化、理解の深化が期待される。
- **Observation Learning:** 観察学習とは、学習者は自分で問題を解決する体験を通じてのみならず、他者の問題解決行動や振る舞いの観察を通じて間接的にも知識やスキルを獲得しうることを指摘したものである。他者の環境への反応の仕方を観察することによってルールが学習され、後に類似した状況に遭遇した際、獲得したルールが適用される。これは、他者の振る舞いを模倣しようとする意図していない場合にも生じる。
- **Cognitive Apprenticeship:** Cognitive Apprenticeship は、他者が新しい知識やスキルを現実的問題に適用する過程を学習者が観察し、ガイドを受けながら実際に自分で行ってみることによって新しいスキルを獲得していく過程である。ここでの課題は、現実的な文脈を提供するものであり、複雑なものへと徐々に変化していくよう準備されることが重要であるとされる。
- **Legitimate Peripheral Participation:** 正統的周辺参加では、あるコミュニティにお

ける新参者としての学習者が、コミュニティのメンバとともに問題解決し、課題を達成していく過程を通じて関連する知識やスキルを獲得し、やがてコミュニティの十全的参加者へと成長していく過程である。そこでは、学習者はコミュニティにおける活動を通して、現実的な文脈での問題を解決する。自分の持つ知識やスキルを実践することによって、他メンバからのフィードバックを得て、さらに洗練していくことが期待される。

- **Self-regulated Learning:** 学習者が現在行っている課題の内容とは別に、現在自分が採用している戦略が効果的か、またプラン通りに実行されているかなど、認知的ゴール達成に関する評価を行う。
- **Sociocultural Theory:** 学習者には、個人で実行可能な領域と実行不可能な領域があり、実行不可能な領域の中には、他者の援助を得れば実行可能な領域が存在する。つまり、発達段階、知識獲得段階の近い学習者同士をペアにして相互作用させることによって、適した領域の機能が他者との間で実現され、それが学習者に内化される。
- **Cognitive Flexibility Theory:** ある事象や知識に関する多角的な見方を提供することは学習者のリフレクション思考を喚起し、自己モニタリングや知識構造の再構築を促す。
- **Distributed Cognition:** 既存知識が一様でない個々のメンバが協力して課題にあたることによって、知識が共有され、集団としての認知が形成され問題が解決される。このとき、学習者は集団として共有された知識を内化していく。

1章で、「学習にはそのプロセスが含まれる」と述べたように、協調学習では学習プロセスに重きを置いた学習である。本研究で扱う協調プログラミング学習支援では、最終的な成果（例えば、プログラムのソースコード）を出すための支援をすることはもちろん重要である。それだけでなく、例えば、学習者間の相互作用をコンピュータ上に蓄積し、それに応じて、学習者グループに対する学習支援を行うこと等、学習者間のインタラクションについても考慮した支援環境の開発が望まれる。

2.1.3 プログラミング学習支援

既存研究におけるプログラミング学習支援は、主にプログラミング初学者を対象とした支援手法が提案されている。

今泉らは、初学者にとってプログラムの実行過程を理解することが難しいと述べており、その原因として、初学者は制御構造、関数を含んだプログラムの適切な動作イメージを作成できていないということに着目し、プログラミング初学者に対し、制御構造と関数の動作を理解することを支援するツールを開発した[今泉, 2010]。制御構造、関数を含んだプロ

プログラミング支援は他にも多くある。喜多らは、プログラムの制御構造に着目したビジュアルライザとして *Avis* を開発している。*Avis* は学習者が記述したプログラムから、プログラムのフローチャートと、取りうる実行経路の一覧をフローチャートとして出力する[喜多, 2005]。新開らは、プログラムのアルゴリズムを逐次、選択、反復の制御構造を用いて、疑似言語で記述し、疑似言語で記述されたプログラムのステップ実行を行うことができるツールを開発している[新開, 2008]。*Terada* は、関数の動作の理解を目的としツールとして、*ETV* を開発している。*ETV* はソースコードの行単位の実行を行い、次に実行される行をハイライトする。プログラムを実行している際に関数呼び出しが行われた際、現在のソースコード表示画面の横に、新たにソースコード表示画面を作成する。関数呼び出しを行ったソースコード画面を残したまま、新しく作成した画面で実行行の表示を行うようにする。この機能を用いることで、ユーザは関数呼び出しの経路を一覧することができ、関数の呼び出しが終了したあと、どの行に戻ってくるかと言うことを理解することができる[*Terada*, 2005]。松田らは、再帰関数の動作の理解に注力したツールを開発している。このツールではユーザにプログラムの部品を組み合わせることで再帰関数を表現させている。ツールは作成したプログラムにおいて、変数の値がどのように変化するかをアニメーションで表現する。変数がどのように変化していくかという履歴を一覧することができるため、再帰呼び出し毎に変数の値がどのように変わっていくのかを理解させることが可能である[松田, 1997]。

プログラミングに関する問題に取り組むことで、プログラミング学習支援をする手法も提案されている。梶田らは、流れ図によるアルゴリズムの理解を含めて学習できる機能をもつシステムを開発している。このシステムでは、穴埋めの個数と箇所を変化できるように自動生成された穴埋め型問題を学習することや、アルゴリズムの流れ図によるプログラミング技法を学習するサポート環境、ならびに複数言語プログラムの実行が可能になるテスト環境を提供している[梶田, 2009]。内田は、ソースコードから作成した穴埋め式のドリルを自動作成し、*Web* を利用して解答が即座に自動採点されるシステムを開発している[内田, 2007]。

他の支援手法として、倉田らは、*C* 言語プログラミング授業のために、実行テストによるプログラム判定をサーバ側で行うシステムの開発を行っている[倉田, 2007] [富永, 2008]。阿部は、プログラミング教育において、*C* 言語を教育するのではなく、*C* 言語を利用してプログラミングを教える場合の *C* 言語が教えにくいものであるということから、初心者向けの *BasicC* というものを提案している[阿部, 2009]。飯田らは、アルゴリズム的な思考法をどのように教えるのかについての試みを行っている。すなわち、プログラミングのための前の段階としてのアルゴリズムでなく、より幅広く物事を分析し、表現できるようにすることを目指している[飯田, 2008]。

本項で挙げたプログラミング学習支援に関連する研究では、協調学習の側面については考慮されていない。本研究で扱う協調プログラミング学習支援では、前述のようにプログ

プログラミング学習の目的に応じた支援環境を開発することが重要であるが、それだけでなく、2.1.2 項で述べた協調学習の側面についても考慮される必要がある。

2.1.4 協調プログラミング学習支援

プログラミング学習支援において、「協調学習の側面を考慮した支援手法」を提案した既存研究がある。

(1) 個人学習からグループ学習への拡張

多くの既存研究では、個人単位で学習する内容をグループ単位で学習するようにした実践が行われている。新開らは、大学の一斉授業を基本として、学習者の理解度を把握しながら C 言語プログラム教育を行うため、グループ活動、他者評価などを取り入れた授業展開を行うことを提案している。特に、プログラム教育の目的であるアルゴリズムを組み立て、プログラムを作成する能力を育成するため、プログラム作成プロセスを重視した指導を行い、その学習効果を明らかにしている[新開, 2009]。山本らは、問題解決的な学習題材として、ロボット制御のプログラムの作成を初等教育に取り入れた。児童の興味・関心を考慮し、ゲーム的活動を取り入れた学習課題を設定し、それらの効果を実験授業で検証した。個別学習になりがちなプログラム学習に共同（グループ）学習を取り入れることで、話し合い活動を通して、作戦を考え、役割分担を行うなど、1つの課題遂行のための様々な知識や技能の習得も可能である。本実践の結果、児童は意欲的に学習に取り組み、課題を遂行する姿が確認できたと述べている[山本, 2006]。黎らは、大学でのプログラミング教育の中のグループプログラミングを中心に、グループプログラミングに関する調査を行っている。調査では、システム開発における基本知識を説明して簡単な仕様書例を提供して2週間に1回システム開発作業報告させる授業と、システム開発の基本知識と詳細なシステム開発手順を説明して、開発手順に合わせて毎週1回仕様書を提出する授業の2つの授業について、ビデオとテープレコーダを用いてグループ相談の様子を観察している。調査の分析からグループプログラミングに対する教育支援システムを提案し、インターネットを利用したシステムを開発した。システムでは、プログラム設計とプログラムの文書化に重点をおき、グループでシステム開発することによって、システム開発に関する知識を獲得し、システム設計の方法、仕様書の書き方などを自然に身につけることができると述べている[黎, 1998]。

(2) 協調学習の有効性を示唆する理論的な背景や現象に即した学習

2.1.2 項で述べた協調学習の有効性を示唆する8つの理論的な背景や現象を踏まえると、Peer Tutoring, Observation Learning を中心とした協調プログラミング学習を実現している既存研究がある。

Peer Tutoring を踏まえた研究では、ある学習者が作成したプログラム等を、保有する知識が等しい学習者（ピア）がレビューするピア・レビューが実践されている。例えば、生田目は、大学の従来の一斉授業を維持し、一部にグループ学習を組み合わせる方式を提案している。この方式では、数名の学生でグループを編成し、グループ内で相互に教えあい、自ら不足している知識を補強しながら課題に取り組む。特にフローチャートについては、学生同士による評価（ピア・レビュー）を取り入れている。この結果、(1)お互いに教えあうことでプログラミングやフローチャートの理解が大幅に向上した、(2)評価することによって、プログラミングの良い具体例を見ることができた、(3)レビューの結果、フローチャートの誤りが発見できたと述べている[生田目, 2004]。北らは、ピア・レビューに基づく学習を支援するための授業支援ツールを開発し、ユーザ評価を行っている。開発したツールでは、ソースプログラムの閲覧・編集機能、チャット機能、コンパイラと連携できる機能を有している。このツールを用いることにより、自分と相手のソースプログラムを比較し、状況に応じてチャット機能を使って相談をしながらソースファイルを編集し、課題を完成させていくことができる。評価実験の結果より、ツール導入によって「エラーの発見が容易になった」、「自分が友人をよく教えた」、「自分が友人からよく教えられた」などのアンケート回答が多く、その結果「理解が深められた」、「協力して課題に取り組めたなどの意見が多く挙げられている[北, 2007]。

Observation Learning を踏まえた研究では、他学習者が作成したプログラムを観察しながら、自身のプログラムを作成するという実践が行われている。例えば、谷口らは、ネットワークを利用してプログラムの作成過程における情報を学生同士で共有することができる Wiki システム Swiki を開発している。Swiki では、電子掲示板の機能に加えて、Web コンテンツの追加と編集ならびにリンクの追加を複数の人がお互いに自由にできる。システムにアクセスすることで、他の学生のプログラムの概略と進捗状況を確認することができる。これにより、プログラミング学習において作品の作成過程における情報を学生が共有することにより、協調的かつ効果的な学習が行われたと述べている[谷口, 2005]。

他にも、プログラミング経験者がこれから学習を行う学習者にプログラミングを教えるという形で、Cognitive Apprenticeship が実現されている例がある。皆月らは、非情報系特有の分野に対応した実践的テーマを創出させてシステムを作りこんでいく授業構成を提案している。システムの作りこみ支援では、過年度の学生の情報や知識リソースが活用可能なしくみを構築し、作りこみを促進させる組織形成手法で作りこみグループを組織化して学生間相互評価を実施している。成果として、過年度の学生のミニ講義による複数年度の学生の技能伝承などを通して、非情報系学生に対してプログラミングに興味関心が向けられて「できる」から「(実際に) つくる」ための構想力の育成を実現したことを挙げている。また、3年間にわたる授業実践では、学生が主体的に授業や課題に取り組む姿勢がみられ、授業から派生した新たな課外活動につながる成果を示している[皆月, 2009]。

また、学生の質問や活動報告に教授者が回答する形で、Cognitive Flexibility Theory が

実現されている例がある。中尾らは、授業形態・内容に合わせて必要な機能を利用できる e-learning 学習支援システムを開発し、そのシステムに用意されている協調学習支援ツールを利用してプログラミング演習を実践している。ツールとして Wiki(PukiWiki)を利用し、伝達事項や Q&A のページとして利用する掲示板ページと、毎回の授業時間内および時間外にどのような作業をしたかを記述させる作業報告ページを用意している。協調学習を利用したことで、学習者の状況が把握しやすくなり、個別対応を丁寧に行うことができたと述べている。また、学習者が自発的に学習方法を工夫する様子が見られたと述べている[中尾, 2006].

(3) ペアプログラミングによるプログラミング学習

協調プログラミング学習としてペアプログラミングを導入する研究も多くある。ペアプログラミングは Beck ら[Beck, 1999]によって提唱されたエクストリーム・プログラミング(XP)と呼ばれるソフトウェア開発手法の 1 つのコンポーネントである。ペアプログラミングでは、1 人が Driver と呼ばれる役割として実際のコードを書き、もう 1 人は Navigator と呼ばれる役割として、Driver をチェックしながらナビゲートし、この役割を随時交代しながら作業を進める。これによって、問題解決に要する時間を短縮させる、常にコードレビューを行うことができる等の効果が得られる。

ペアプログラミングによるプログラム作成は協調作業であり、また、プログラミングに関する専門知識の共有および拡張という点でも有効という指摘がある[Wray, 2010]。また Preston は、ペアプログラミングと協調学習の関連を「Common task or learning activity suitable for group work」、「Small group learning」、「Cooperative behavior」、「Interdependence(or positive interdependence)」、「Individual accountability and responsibility」の 5 つの観点で示し、ペアプログラミングを利用した協調学習が有効であることを示している[Preston, 2006].

このペアプログラミングを用いたプログラミング学習が行われている。特に、プログラミング導入教育におけるペアプログラミングの利用では、個人でプログラミングをするときと比較した場合に、プログラムの質が向上する、課題解決までの時間が短縮する等、その有効性が報告されている[McDowell, 2002][Nagappan, 2003][Hanks, 2004][Rountree, 2005]。プログラミング導入教育を対象とした研究ではないが、作業中のペア同士のコミュニケーションを分析している研究がある。Chen らは、作業中のペア同士の発話内容と発話時の状況を書き下し、発話内容から Driver と Navigator には精神的な距離があること、作業中に何らかのコミュニケーション支援が必要であることを示している[Chen, 2007]。Chong らも、作業中のペア同士の発話内容と発話時の状況を書き下し、ペアが所有する専門的な知識が作業中のインタラクションに影響する可能性があること、キーボードの操作権(Keyboard control)がペアの意思決定に影響する可能性があることを示している[Chong, 2007]。Bryant らは、作業中の発話プロトコル分析を行って Driver および Navigator の発

話内容の分布を調査し、両者の分布に差はなく、同じレベルで相互作用していることを示している[Bryant, 2008].

2.2 本研究の位置づけ

本研究では、協調プログラミング学習支援に着目している。本節ではまず既存の協調プログラミング学習支援についてまとめ、問題点を明らかにする。その後、問題点の解決に向けて何をすべきかということをする。

2.2.1 まとめ方針

協調プログラミング学習を含む協調作業を支援するグループウェアは空間・時間の 2 つの側面で分類されるのが一般的とされている[農林水産省, 1996]。グループウェアとは企業等の組織の中でのグループ作業を支援するソフトウェアのことを指す。グループウェアはそれぞれ、グループメンバが「同じ場所で作業をするのか（対面型/分散型）」、「同時に作業するのか（同期（リアルタイム）型/非同期（蓄積）型）」という側面で分類される。

表 2-1 はグループウェアの分類を示したものである。対面同期型（分類 a）は主に会議室内での会議支援が応用の中心である。一ヶ所に集まって作業を行う場合の進行・分担の支援に応用できる。分散同期型（分類 b）は物理的に離れた場所での同時協調作業を支援するものである。大区画の圃場や分散した圃場での作業同士の情報交換、あるいは作業グループの統括や適切なアドバイスの提供を遠隔地からでも行えるようにすることが考えられる。分散非同期型（分類 c）は電子メールや電子掲示板をベースにした形態のシステムが中心で、グループ間の情報の蓄積・共有や空間的にも時間的にも分散した作業の調整支援を目的としている。単に電子メールや掲示板のようなメッセージ交換の機能を提供するだけでなく、その上で行われるグループワークの構造的特徴に即したシステム設計が重要となる[農林水産省, 1996].

表 2-1 グループウェアの分類 [農林水産省, 1996]

	同期	非同期
対面	分類 a ・電子会議室	
分散	分類 b ・遠隔会議システム ・グループエディタ	分類 c ・プロジェクト管理 ・情報共有

協調作業においては、その作業に参加する人数（グループサイズ）により得られる効果が異なることがある。グループサイズについて考察している中野の研究[中野, 2003]では、話し合いの場において、2人だと話がきちんと伝えあえ、3人だと相乗効果が生まれやすく、4人になるとアットホームでかつ多様性が生まれ、5～6人を超えると関与度が低い者が出てくるということを指摘している。この指摘を踏まえ、表 2-1 の分類を改良した。

表 2-2 は表 2-1 の分類について、さらにグループサイズについて分類したものである。グループサイズの分類は、2人グループ、小規模グループ（3～4人）、大規模グループ（5人以上）の3つとした。2.1.4項で挙げた既存の協調プログラミング学習を実践している研究について、表 2-2 にまとめることにした。

表 2-2 本研究におけるグループウェアの分類

	同期			非同期		
グループ サイズ	2人	3～4人	5人～	2人	3～4人	5人～
対面				/		
分散						

2.2.2 既存研究のまとめ

2.1.4項で挙げた協調プログラミング学習を実践している研究について、表 2-2 で示した分類に従ってまとめたものを表 2-3 に示す。

協調プログラミング学習は大学の講義等の中で実践している研究が多く、ほとんどの研究において対面環境における学習を支援している。対面同期環境で、2人グループで協調プログラミング学習を行った実践では、ある学習者が作成したプログラムやフローチャートを他の学習者がレビューするピア・レビュー[生田目, 2004][北, 2007]や、2人1組になってプログラミングを行うペアプログラミング[McDowell, 2002][Nagappan, 2003][Hanks, 2004][Rountree, 2005][Chen, 2007][Chong, 2007][Bryant, 2008]が行われている。対面同期環境で、3～4人グループあるいは5人以上のグループで協調プログラミング学習を行った実践では、個人単位で学習する内容についてグループ内で話し合うことで、グループ単位で学習するようにしている。

分散環境における学習を支援している研究について、分散非同期環境で、3～4人グループあるいは5人以上のグループで協調プログラミング学習を行った実践では、Wiki やデータベースにプログラムの設計文書等を蓄積し、それを閲覧できるようにすることによって、距離的に離れている学習者同士で学習内容を共有することを可能としている。

表 2-3 既存研究のまとめ

グループ サイズ	同期			非同期		
	2人	3~4人	5人~	2人	3~4人	5人~
対面	[生田目, 2004] [北, 2007] [McDowell, 2002] [Nagappan, 2003] [Hanks, 2004] [Rountree, 2005] [Chen, 2007] [Chong, 2007] [Bryant, 2008]	[新開, 2009] [山本, 2006]	[皆月, 2009]	/		
分散					[黎, 1998]	[谷口, 2005] [中尾, 2006]

2.2.3 既存研究の問題点

既存研究のまとめを踏まえると、協調プログラミング学習支援について分析・考察する上で、次の2点に問題があると考えられる。

(1) 分散環境における協調プログラミング学習支援例の少なさ

1章で述べたように、情報技術の発達により、学習者が同じ場所の学習環境に存在するような対面環境のみではなく、距離的に離れた場所に存在する学習者同士でやり取りするような分散環境での学習も可能となっている今日では、分散環境での学習も普及していくと考えられる。林の報告[林, 2004]によれば、地理的に分散な環境において、大規模の人数で学習を進めていくようなeラーニングシステムが今後注目されるとの指摘もある。今後、協調プログラミング支援環境を開発していく上では、既存の分散環境における学習も考慮する必要があると考えられるが、現状ではその実践例が少ない。特に表2-3で示した分散環境における学習を扱った研究では、アンケート調査から得られた結果をまとめる等の定性的な評価で研究を結論づけており、この結果だけでは、十分な分析・考察ができない。

(2) 協調プログラミング学習における学習不振者に関する考察の少なさ

表2-3で挙げたすべての研究において、個人で学習するよりも、「エラーが少なくなった」、「成績が向上した」など、協調プログラミング学習を支持する結論を述べている。しかし、

プログラミング学習がうまくいかなかった学習不振者に関する考察は少ない。1章で述べたように、プログラム言語の文法や簡単な例題を理解することができても、実際にプログラムを作成するとなったとき、プログラムを組むことができない[新開, 2009]等の問題が生じ、学業不振になってしまう学習者もいる。また、このような学習者を学習過程の早い段階で見つけ出し、弱点補強のための練習問題の提供や、学習方法に関するアドバイスの提供等、具体的な支援を行うことが望まれている[匂坂, 2009]。協調プログラミング学習を支援していくには、学習不振者に対してどのように支援するかについても考慮していく必要があるし、そのためには既存の支援において、どのように協調プログラミング学習がうまくいかなかったのかについて調査・分析する必要がある。

2.2.4 問題点の解決

前項の問題点を解決するため、本研究では分散環境における協調プログラミング学習と対面環境における協調プログラミング学習を実施する。これにより問題点を解決するとともに、次の2点を新たに明らかにする。

(1) 分散環境で大規模なグループにおける協調プログラミング学習に対する支援効果

表 2-3 で示した分散環境における学習を扱った研究では、アンケート調査から得られた結果をまとめる等の定性的な評価で研究を結論づけているため、本研究では定量的な評価を行い、支援効果等を考察する。これにより、分散環境における協調プログラミング学習支援例を新たに示すことになり、2.2.3 項で述べた1つ目の問題点の解決につなげる。

(2) プログラミング中の学習不振者に関する考察

表 2-3 に示した研究では、プログラミング学習がうまくいかなかった学習不振者に関する考察が十分ではない。そこで、本研究で協調プログラミング学習を実施し、分散環境および対面環境における協調プログラミング学習それぞれの学習不振者について考察する。これにより、2.2.3 項で述べた2つ目の問題点の解決につなげる。

本研究で実践した協調プログラミング学習が、表 2-3 で示した分類のどこにあたるかについて、表 2-4 に示す。本研究では、分散環境における協調プログラミング学習として、分散非同期環境で利用する作問学習支援システムにおいて実施したプログラミング学習を調査・分析した[平井, 2010]。また、対面環境における協調プログラミング学習として、対面環境で実施したペアプログラミング学習を分析した[平井, 2012]。このような実践を設定した経緯やそれぞれの学習環境については、第3章、第4章の冒頭で説明する。その後の第5章において、既存研究の知見と本研究の知見を踏まえ、分散環境における協調プログラミング学習支援と対面環境における協調プログラミング学習支援という観点で検討を行う。

また、今後の協調プログラミング学習支援についてその展望を述べる。

表 2-4 本研究で実践した学習の分類

グループ サイズ	同期			非同期		
	2人	3~4人	5人~	2人	3~4人	5人~
対面	[生田目, 2004] [北, 2007] [McDowell, 2002] [Nagappan, 2003] [Hanks, 2004] [Rountree, 2005] [Chen, 2007] [Chong, 2007] [Bryant, 2008] <u>[平井, 2012]</u>	[新開, 2009] [山本, 2006]	[皆月, 2009]	/		
分散					[黎, 1998]	[谷口, 2005] [中尾, 2006] <u>[平井, 2010]</u>

第3章 分散環境における協調プログラミング学習の分析

本章では、分散環境における協調プログラミング学習の分析として、分散非同期環境で利用する作問学習支援システムにおいて実施されたプログラミング学習を調査・分析した結果を報告する。

2.1.3 項で述べたように、プログラミングに関する問題に取り組むことで、プログラミング学習支援をする手法が提案されており[梶田, 2009][内田, 2007]、一定の効果が得られている。この手法では、個人学習を行うことを想定している。本研究ではこの手法に協調性を取り入れることで協調的なプログラミング学習ができると考え、複数人で問題に取り組むことができる作問学習の手法を取り入れた。

本研究では、作問学習支援システム *Concerto III* を利用する。*Concerto III* では、(1)ある学習者が学習内容に対する問題を作成し、(2)作成した学習者とは別の学習者がその問題に解答し、(3)解答した学習者がその問題に対する評価を行う。その後、(4)作成した学習者や解答と評価をした学習者を交えてその問題の内容に関する議論を繰り返し行い、(5)作成した学習者は必要に応じて問題を修正する、という利用プロセスを想定している。このプロセスに従って、プログラム問題の作成、評価、修正を行うことにより、複数人で1つの問題を作成することができる。本研究ではこの活動を協調プログラミング学習とした。この活動を分散非同期環境において大規模の人数で行うことにより、分散環境で大規模なグループにおける協調プログラミング学習を行うことを実現した。

3.1 研究概要

作問とは、問題を作るという活動である。この活動は、問題を解く活動よりも高度に知的な活動であり、学習者の問題解決能力の向上に寄与するといわれている[横山, 2005]。問題を作るという行為そのものは、ある特定の要件を満たした問題を効率的に解くための一連の操作とそれを適用するための要件判定までの操作を含めた一塊の知識の再構成を促す作業であり、関係的理解の促進につながると期待できる[平嶋, 2005b]。また、この際、原理原則から再構成するのではなく、自身の行っている問題解決に対する振り返りとして行われることから、学習者に問題解決や知識をメタ認知させるための方法として位置付けることができる[平嶋, 2005b]。

近年では作問による学習方法に加え、他の学習者が作成した問題に対する評価や作成さ

れた問題に関して議論するという学習方法も取り入れられている。学習者が作成した問題を共有し、他の学習者がそれらを解くことや問題に対する質疑応答を通じて、学習内容に関する理解力が向上することが期待できると述べられている[Yu, 2005][高木, 2007]。また、学習者が問題に対する評価を行うためには、問題文の適正さや解答・解説の正しさを客観的かつ批判的にチェックする必要がある、この活動には認知的プロセスが含まれると言われている。さらに、他の学習者とインタラクションを行うことが認知的発達につながるとも言われている[Yu, 2005]。

これらの背景のもと、モンサクン[横山, 2007]やQPPA[Yu, 2005]をはじめとして、多くの作問学習支援システムが開発されるようになり、筆者らも分散環境で作問学習が可能な学習支援システム **Concerto** を開発した[Hirai, 2007][平井, 2008]。Concerto は作問、解答、問題に対する評価、問題について質疑応答を行う機能を提供する。Concerto は Web を用いた学習支援システムであるため、学習者は Web を利用できる環境が整っていれば、システムをいつでもどこでも利用可能である。

作問学習支援システムの分析において問題を作成することに対する効果の分析は行われてきた。しかし近年開発されている作問学習支援システムでは作問以外の活動も多数行われており、それらの重要性は述べられているが、分析による裏づけは多くない。本章で触れるが、著者らもこれまで **Concerto** を改良した **Concerto III** を大学の講義で利用し、システムに蓄積されたデータやシステム利用前後のプレテストとポストテスト成績を用いて分析を行った。その結果、問題を作成することに対する効果以外にも、誤字・脱字が少なく、解答・解説が間違っていない正確性の高い問題を作成することの効果[平井, 2010]、作成された問題を評価することによる効果[Hirai, 2009]を示した。

これにより、問題を作成すること、正確性の高い問題を作成すること、作成された問題を評価することに効果があると示された。しかし、人間を対象とした実験では効果の要因は複雑で分離することができない[赤堀, 2002]といわれるように、これらの効果の要因はそれぞれ単独ではなく、複合的に発生している可能性がある。多くの既存研究では作問学習の効果の要因として 1 つ、あるいは多くとも 2 つを組み合わせたものまでしか報告しておらず、それぞれの活動が作問学習において有効であることは報告しているが、それらの関係は示されていない。

本章では、作問学習支援システム **Concerto III** の適用実験結果をもとに、システムの適用効果に関する分析を行った。その結果、問題を多く作成している学習者はそうでない学習者よりも、システム適用前に行われたプレテストから適用後に行われたポストテストへの成績の伸びが大きかったことが分かった。また、システム内で学習者が評価した問題の正確性評価を用いて分析を行った結果、正確性の高い問題を作成する学習者はそうでない学習者よりもプレテストからポストテストへの成績の伸びが大きかったことが分かった。さらに、システムの導入により、システムが学習者に対して学習のきっかけを与えていたことが分かった。

また、作問学習支援システムを利用した作問活動について相関ルールマイニングを用いて学習効果の分析を行い、作問学習における学習活動と成績の関係を調査した。その結果、作問学習支援システムを利用することにおいて、システムを積極的に利用すること、問題を解答すること、作成する問題の正確性も成績に関係していると考えることができた。そして、問題に解答することが成績向上に大きく関係していること、正確性の低い問題を作成することが成績低下に大きく関係していることが新たに分かった。

3.2 関連研究

学習者による作問学習とその効果は平嶋らの研究[平嶋, 2001]や中野らの研究[Nakano, 1999][中野, 2000][中野, 2002]をはじめとして、様々な形で実践研究され、その結果が報告されている。本節では関連研究との比較を行う。

3.2.1 システムの機能・運用・評価

本研究では、学習者による作問、学習者による問題の評価、学習者による問題に関する議論を支援するシステムに着目しているため、これらを扱っている研究の中での位置づけを述べる。本研究と関連研究を比較し、結果を表 3-1 にまとめた。

作問形式について、本研究では学習者が自由な発想で作問することができるように、作問形式として多肢選択式のみでなく、自由記述による形式も提供している。問題に対する評価や議論について、Yu らのシステムは評価コメントを記述することにより作問者にフィードバックを与えることは可能であるが、作問者とそれ以外の学習者による双方向のコミュニケーションを支援してはいない。本研究で開発したシステムで提供している「作問要求機能」、「採点機能」、「問題の正確性を測る指標」について、「作問要求機能」、「問題の正確性を測る指標」は本研究でのみ提供している。「採点機能」は菅原らのシステムでも提供している。本研究で「問題の正確性を測るための指標」として用いているポイント制やランキング表示は、高木らの研究ではこれらを「学習者グループ間で競争させるため」に導入している[高木, 2007]。

学習者が主体的に作問を行うという点では全ての研究で同じであるが、高木ら、菅原ら[菅原, 2007]、渡辺の研究[渡辺, 2005]ではグループを単位として活動を行っている。グループ単位で活動を行うと、グループ内の特定の学生しか作問学習活動を行わなくなってしまう可能性を考慮して、本研究では個人を単位として作問学習活動を行わせることとしている。また、本研究では授業時間外の分散非同期環境でも作問学習活動を行うことも想定している。

システムの評価について、表 3-1 で挙げた研究では、システムの利用アンケートやシステ

ム（機能）の利用回数等を用いて評価を行っているが[Denny, 2008][Hayashi, 2008][高木, 2007], 作成された問題の正確性を利用した分析は行われていない。

表 3-1 本研究と関連研究の比較

	高木ら	Yu ら	菅原ら	Hayashi ら	渡辺	Denny ら	本研究
作問形式	多肢選択 (テキスト ベース)	多肢選択 (テキスト と図)	自由記述 (手書き作 問可)	多肢選択 (テキストと 図)	自由記述 (テキスト)	多肢選択	多肢選択・自 由記述(テキ ストと図)
問題評価	○	○	○	—	○	○	○
問題に関する議論	○	—	○	△(不具合の 報告目的)	○	○	○
問題の正確性を測る指標	—	—	—	—	—	—	・ポイント ・ランキング
活動単位	グループ	個人	グループ	個人	グループ	個人	個人
適用環境	対面同期	対面同期	対面同期	分散非同期	分散非同期	分散非同期	分散非同期
モチベーション向上支援	・テスト問題へ利用 ・ポイント ・成績表示 ・ランキング表示	—	—	・成績表示 ・ランキング表示	テスト問題へ 利用	—	最終成績への 反映
教授者の介入	問題レビュー	—	評価や採点 対象問題を選択	—	—	—	議論時のコメント(学習者の要請時)
システムの有効性を評価するための材料	・システム 利用回数 ・利用アンケート ・成績情報 ・システムの蓄積データ	利用アンケート	・活動時間 (紙上とシステム上の比較) ・利用アンケート	システムの利用回数	利用アンケート	・システム利用回数 ・成績データ	・システム利用回数 ・アンケート ・成績情報 ・システム蓄積データ ・問題の正確性情報

3.2.2 作問学習支援システムにおける学習者の活動分析

作問学習支援システムにおける学習者の活動分析やその効果を検証する際、当初は、実際にシステムを利用した学習者が回答したアンケートを利用する定性的な分析が行われた。アンケートの分析結果から、例えば、次のようなことが報告された。

- 単に問題を作成させるだけではなく、問題作成の過程でコメントを投稿しあうグループレビューを行うことにより、学生同士のインタラクティブ性が向上し、その結果学生の学習意欲が向上することが示唆された。また、問題を作成することによる理解度の向上と、学生同士で作成した問題を解きあう有効性が示唆された。[高木, 2007]
- 多桁減算を対象としたシステムを用いた学習活動は学習として有用であると考えられた。[東本, 2007]

他にも、問題を作成することに有効性があるという報告がいくつかある[菅原, 2007][Yu, 2005][Hayashi, 2008][渡辺, 2005]。

近年、定量的な分析を行う研究が多く報告されており、作問学習支援システムにおける学習者の活動分析は、システムに蓄積されたログを用いられるようになった。また、作問学習の効果を検証する際は、ログの分析と合わせて、システム利用前のプレテストやシステム利用後のポストテストの成績を用いることも多い。例えば、次の2点のように報告された。

- プレテストの成績順に学習者を2つの学習者群に分け、それぞれのグループにおいて、問題を多く作成したグループとそうでないグループに分けて、ポストテストの成績を分析した。その結果、プレテストで成績の悪い学習者群で問題を多く作成したグループでは、プレテストとポストテストの成績に有意差があり、システム利用の効果が見られた。[横山, 2007]
- プレテストの成績順に学習者を4つの学習者群に分け、それぞれのグループにおいて、システムをよく利用した学習者グループ(M)とそうでない学習者グループ(L)に分ける。ポストテストについて、グループMとグループLで比較した結果、プレテストで最も成績が良かった学習者群と、最も成績が悪かった学習者群で、グループMとグループLに有意差があった。このことから、元々成績の良かった、あるいは悪かった学習者にシステム利用の効果があった。[Denny, 2008]

他にも同様の分析手法を用いた例が報告されている[中野, 2002][高木, 2007][東本, 2007][岡本, 1996][Hirashima, 2008]。

対象とする学習者、作問する内容や形式により得られる効果は異なる。本項で述べた従来の研究では、

- 「問題を作成することによる理解度の向上」、「問題を解きあうことの有効性」というような単独要因
- 「成績の悪い学習者群で問題を多く作成したグループではシステム利用の効果が見ら

れた」というような2つの複合要因による学習効果しか検討されておらず、作問学習における様々な活動がどのような関係で学習効果に影響しているのかということは検討されていない。

3.2.3 データマイニング手法

作問学習支援システムも含む一般的なLMS (Learning Management System) では、演習問題への解答履歴や電子掲示板への投稿履歴などがデータベースに蓄積される[植野, 2007]。データベースに蓄積される学習者の膨大な学習履歴データから、重要な知識やデータを発見・抽出するために、データマイニング手法が多く利用されている。

データマイニング手法は、AGRAWALらが提唱した相関ルール[Agraval, 1993]をはじめとして、決定木、サポートベクターマシン、クラスタリング等がある。重要な知識やデータを多く発見できることが望ましいが、それが膨大になってしまうと解釈することが困難になる。本研究では、閾値(後述)を設定することにより、抽出するルール数を調節することができる相関ルールマイニングを利用する。

3.3 作問学習支援システム

本章では、本研究で開発したシステムの利用プロセスとその利用プロセスを踏まえて開発したシステム ConcertoIIIの概要を述べる。

3.3.1 システムの利用プロセス

図3-1にシステムの利用プロセスを示す。学習者はシステムで、①学習内容に対する問題を作成し、②作成した学習者とは別の学習者がその問題に解答し、③解答した学習者がその問題に対する評価を行う。その後、④作成した学習者や解答と評価をした学習者を交えてその問題の内容に関する議論を繰り返し行い、⑤作成した学習者は必要に応じて問題を修正する、というプロセスを繰り返す。また、システムでは学習者が他の学習者(クラス全体)に対して、作問を要求することができる機能(⑥)、作問時における議論支援機能(⑦)、作問要求時における議論支援機能も提供する。

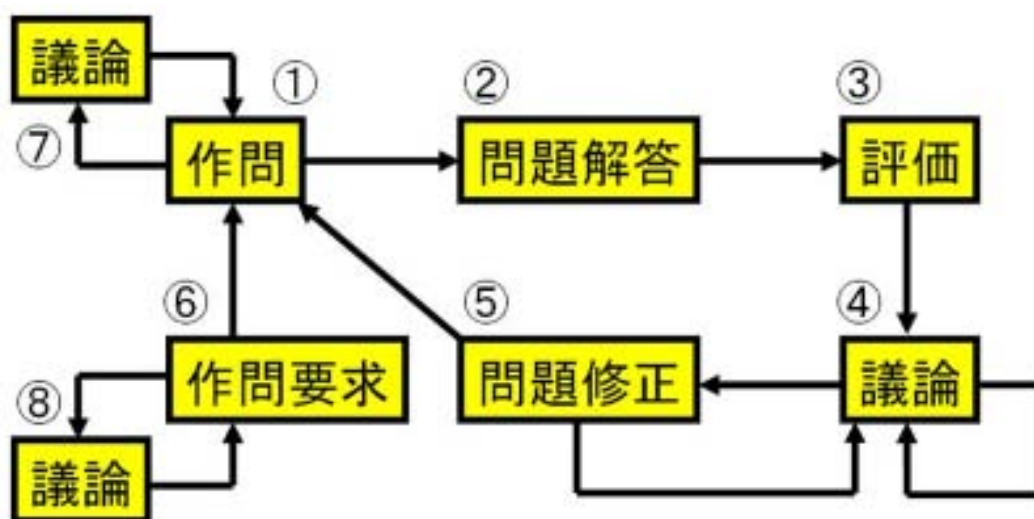


図 3-1 システムの利用プロセス

3.3.2 Concerto III の概要

3.3.1 項で示した利用プロセスをもとに、作問学習支援システム Concerto IIIを開発した。本システムは Java 言語を用いて開発した。ユーザインターフェースに JSP(Java Server Pages)を、Web アプリケーションサーバに Tomcat を、データベースに MySQL を利用した。システム利用の際、特別なインストール作業は必要とせず、Web を利用できるコンピュータであれば、いつでも利用することが可能である。以下に、本システムの主な機能や特徴を述べる。

(1) 作問

学習者が、科目を選択し、その科目の学習内容に対する問題を作成することができる。作問形式として選択式問題、記述式問題の 2 種類から一方を選択し、その形式に応じた問題を作成することができる (図 3-2)。また、図を用いた問題の作成も可能であり、問題登録後に必要に応じて修正することもできる。

(2) 問題解答と問題評価

作成された問題 (自身が作成したものは除く) に解答することができる。また、問題に解答し、問題に対する評価を登録できる。問題に対する評価 (5 段階評価) として、新規性評価・信頼性評価・難易度評価と評価コメントを登録することができる (図 3-3)。なお、各評価項目において教授者が設定した評価基準を示すことが可能である (図 3-3 右側の評価基準ボタンを押すことにより表示)。本研究では、新規性評価、難易度評価、信頼性評価について、次のように 5 段階評価の評価基準を設定した。:

問題作成 (※は必須)

問題作成中に分からないことがあったら → [作問用会議室へ](#)
別ウィンドウで表示されるので、記入中でも記入内容を残したまま利用できます。

※注意:セキュリティの関係で30分間放置するとユーザ情報を無効にするようにしています。
問題・解説を作成する時間が長くなりそうな方は、問題・解説が完成してからこの画面にくと良いかも。

●作問フォーム●	
* 作問者	平井
* 科目	計算機システム概論(2008)
* 想定難易度	(未選択) ↓ →この問題の想定難易度を選択してください。
* 分野	(未選択) ↓
* 問題文	<div style="border: 1px solid gray; height: 40px;"></div>
問題用の図	<div style="border: 1px solid gray; height: 20px;"></div> 登録 削除
* 選択肢	ア: <input style="width: 80%;" type="text"/> イ: <input style="width: 80%;" type="text"/> ウ: <input style="width: 80%;" type="text"/> エ: <input style="width: 80%;" type="text"/>
* 正答	(未選択) ↓
* 解説文	<div style="border: 1px solid gray; height: 40px;"></div>
解説用の図	<div style="border: 1px solid gray; height: 20px;"></div> 登録 削除
* 出題意図	この問題を出題した意図は? <div style="border: 1px solid gray; height: 20px;"></div>
* 参考資料	この問題を作成するために参考にした資料は? →記入例 <div style="border: 1px solid gray; height: 20px;"></div>
工夫点	この問題を作るときに工夫した点は?(この点を評価して欲しい! など) <div style="border: 1px solid gray; height: 20px;"></div>

[問題登録](#)

図 3-2 問題作成インターフェース

■ 新規性評価

1. 同じ問題を前にも見た (数値を変えただけ等)
2. 選択式 or 記述式で同じような問題を見た
3. どこかで見たことがあるような…等の推測でしか判定できない。
4. 答えは同じだが、問題文の書き方は新しい
5. 教科書、インターネット等でも見たことがないぐらい新しい

■ 信頼性評価

1. 問題文に矛盾している箇所がある． または問題に解答することができない（解なし等）
2. 問題に解答することは可能だが， 作問者が示した正答または解説に誤りがある
3. 問題文， 解答， 解説に誤りはないが， 解説をもう少し詳しく書く必要がある
4. 問題文， 解答， 解説に誤りはなく， 解説も詳しく書かれているが， 問題文， 解答， 解説に誤字， 脱字等がある
5. 問題文， 解答， 解説に誤りはなく， 解説も詳しく書かれている

■ 難易度評価

1. 暗記しているだけで解答できる
2. 問題集の基本問題レベル／数をこなすにはちょうど良い
3. 教科書の例題レベル／学習の導入として利用できる
4. 問題集の応用問題レベル／試験に出てもおかしくない
5. 問題集の章末問題レベル／試験には出せないほど難しい

問題評価

↓この問題・解説・問題作成の努力の評価に御協力を！
評価をして「評価登録」ボタンを押して下さい。
問題に関して質問・疑問等があったら会議室に投稿してください。

※答えと解説は画面下部にあります

●問題評価フォーム● *は必須	
* 正誤判定	正解しましたか？ <input checked="" type="radio"/> 正解でした <input type="radio"/> 不正解でした <input type="radio"/> どちらとも言えない
* 新規性評価	この問題形式を見るのは初めて？ 新: <input type="radio"/> 5 <input type="radio"/> 4 <input type="radio"/> 3 <input type="radio"/> 2 <input type="radio"/> 1 : 古 →評価基準
* 難易度評価	あなたにとって、この問題は難しかったですか？ 難: <input type="radio"/> 5 <input type="radio"/> 4 <input type="radio"/> 3 <input type="radio"/> 2 <input type="radio"/> 1 : 易 →評価基準
* 信頼性評価	この問題を問題集に載せても良いですか？ 良: <input type="radio"/> 5 <input type="radio"/> 4 <input type="radio"/> 3 <input type="radio"/> 2 <input type="radio"/> 1 : 悪 →評価基準
* コメント	数値だけでは完全な評価ができないので、さらにコメントを！ 特に、上の点数をつけた理由や工夫点の評価をお願いします。 <div style="border: 1px solid gray; height: 40px; width: 100%;"></div>

評価登録
→この問題に関して会議室で議論

図 3-3 問題評価インタフェース

(3) 掲示板

問題単位での議論が可能なように， 各問題に掲示板を提供している． 議論支援において「教授者のコメント要求」を提供した． これは学習者同士による議論を進めていく際に，

学習者だけでは解決できなくなる場合を考えて提供した。この機能を利用すると、教授者に「コメントの要求がある」という内容のメールが送信される。これにより、学習者は優先的に教授者のコメントを受け取ることが可能となる。

(4) 作問要求

学習者が作問を要求することができる。図 3-4 は要求された内容の一覧画面である。各要求に対して、要求者と要求内容に加え、その要求に応じて作成された問題が関連付けて表示される。また、要求内容に対する質問等ができるように、図 3-1 の⑧に相当する要求掲示板を設置した（図 3-4 右側の「会議室」ボタンを押すことで利用できる）。

作問要求一覧					
新しく要求をする → 作問要求する					
要求日	要求者	要求内容	要求に合う問題の問題番号	各要求に 関する 会議室 ↓	要求に合う 問題にリンク させる ↓
08-07-15	あー	記憶装置の計算系(P184の間3みたいな)の問題作ってください!!	245 →解答する	→会議室	問題番号→ →リンクづけ
08-06-30	平井	(6月10日までに作成されていなかったもので) MIL記号を利用した問題を作成してください。	203 →解答する 232 →解答する	→会議室	問題番号→ →リンクづけ
08-06-25	平井	これは素晴らしいと思った問題をリンク付けてください! (作問要求機能を使ってみよう!)		→会議室	問題番号→ →リンクづけ

図 3-4 作問要求インターフェース

(5) 作問ポイントと貢献ポイント

学習者がこれまでに述べてきた機能を利用することで、その学習者に対して作問ポイントや貢献ポイントを与えるようにした。

作問ポイントは、ある学習者が作成した問題に対して他の学習者から評価があった場合に、その評価値（新規性、問題の正確性、難易度の 5 段階評価）に応じたポイントが作問者に与えられる。この作問ポイントを用いることによって、その学習者が作成した問題は他の学習者からどの程度の評価を受けているかということが分かり、作成された問題の信頼性を量る一つの指標になると考えられる。

貢献ポイントは、システム内で次の活動をしたときに与えられる。:

- 他の学習者が作成した問題を評価する
- BBS にコメントを投稿する
- 作問要求に応えた問題を作成する

この指標を用いることによって、その学習者が他の学習者に対してどのくらい貢献したかが分かる。

これらを用いて、作問ポイントと貢献ポイントに応じた学習者のランキングを表示できるようにした[平井, 2008].

3.4 システム適用実験

本節では、ConcertoIIIの適用実験とその結果について述べる.

3.4.1 実験概要

ConcertoIIIを東京学芸大学教育学部情報教育専攻で開講されている「計算機システム概論」の2008年度および2009年度の受講者に対しシステムの適用実験を行った. この科目は情報教育専攻の学部1年生対象の必修科目であり、コンピュータのしくみを理解することを目的としている. 適用実験後にシステムに対するアンケートへの回答を依頼した.

システムの適用期間は6~7週間であり、この期間内において、受講生がシステムに自由にアクセスできるようにし、システムの全機能を好きなときに好きなタイミングで自由に利用させた. また、授業中はシステムの利用説明のみを行い、その他すべての演習は授業外の時間に行わせた. さらに、教授者から受講生に対して「1人2問は作問し、1回はシステムの全機能を利用してください」という指示を出した. ポイント制について、教授者から受講生に対して作問ポイントや貢献ポイントがどのようなこと(3.3.2項(5))をすると増えるかということを伝えた.

システムを適用したことによる学習効果を検証するため、システム適用前に中間テスト(計算機システム概論の中間試験)、システム適用後に期末テスト(計算機システム概論の最終試験)を実施した.

3.4.2 実験結果

表3-2にConcertoIIIの適用実験終了後のシステムの利用状況を示す. 図3-5にプレテストとポストテストの成績分布を示す. 成績はプレテスト、ポストテストそれぞれについて偏差値に変換している. 中間テストの試験範囲はそれまでに学習した内容(データの内部表現、論理回路)とし、事前にテストを受けるか受けないかの判断は各学習者に任せた. また、期末テストは100点満点であり、試験範囲は全範囲(データの内部表現、論理回路、記憶装置、アセンブラ、コンピュータ史)である.

表3-3にConcerto IIIに登録された問題とその問題に設置されているBBSで行われた議論の例を示す. この例を見ると、この問題に関する議論は、問題の解答・解説に対する質問から始まり、それに対する回答が作問者からあり、質問をした学習者は感謝をしている.

さらに、他の学習者が、作問者が回答した内容とは別の説明をし、この説明に対して、複数の学習者が「分かりやすい」とコメントしており、作問者もこれに同意をしているということが分かる。つまり、この議論は、作問者に対する質問とその回答をベースとして、作問者・質問者とは別の学習者が、異なる観点から説明をした結果、作問者も含めて複数の学習者が問題の解説内容を理解したという流れになっている。

表 3-2 ConcertoⅢの利用状況

年度	2008	2009
公開期間	約 6 週間	約 7 週間
ユーザ登録数	94	77
ログイン数	1384	611
問題作成数	273	94
問題解答数	3815	1340
問題評価数	2561	697
掲示板投稿数	23	9

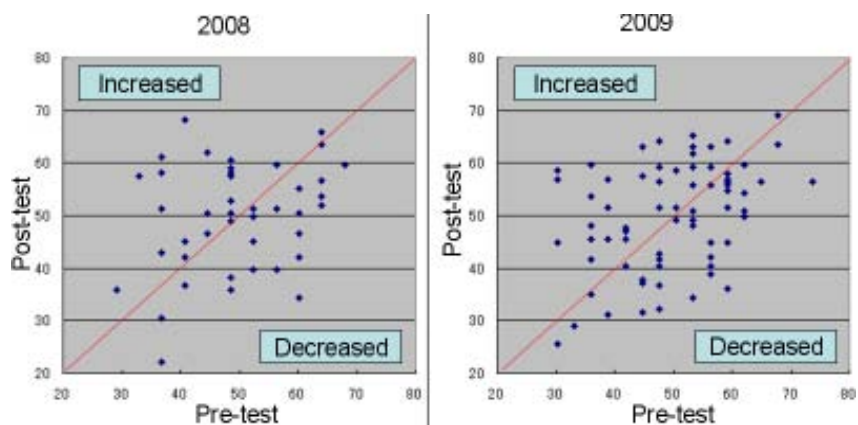


図 3-5 プレテストとポストテストの成績分布。

図中の点は各学習者を表し、斜線よりも上にいる学習者はプレテストからポストテストにかけて成績が向上したとする

表 3-3 問題とその問題に関する議論の例

問題 (左から作問者, 作問日時, 問題文)		
学習者 A	2007/05/30 22:34:39	<p>次のプログラムを正しく実行させるためには[ア],[イ]には何をいれればよいか答えなさい.</p> <pre> MASUO START LAD GR7,[ア] LAD [イ],DQN LD GR2,0,GR7 LD GR3,1,GR7 LD GR4,2,GR7 LD GR5,3,GR7 LD GR6,4,GR7 ST GR2,0,GR1 ST GR3,1,GR1 ST GR4,2,GR1 ST GR5,3,GR1 ST GR6,4,GR1 RET ORG DC 'MASUO' DQN DS 5 END </pre>
解答・解説 (左から作問者, 解説日時, 解答・解説文)		
学習者 A	2007/05/30 22:34:59	<p>[ア]ORG [イ]GR1 ちなみに実行結果として DQN+0 から DQN+4 には MASUO が格納されます.</p> <p>すべて説明するのは困難なので要点だけ説明させていただきます.</p> <p>[ア]について, LAD GR1,ORG では ORG の領域に格納されている値ではなく, ORG 自身,つまり ORG の領域の先頭番地が GR1 に設定されている. LAD はアドレスの設定をし, LD は値の設定をする.</p> <p>[イ]について, ST 命令のところで GR1 をすでに指定しているから.</p>
問題に関する議論 (左から投稿者, 投稿日時, 投稿内容)		
学習者 B	2007/06/11 15:10:01	<p>すみませんが, LD と LAD の違いがまだわからないのでおねがいします.</p>

学習者 A	2007/06/12 19:11:40	<p>さて、LD と LAD の違いがわからないという質問をお受けしましたので浅はかな知識ながら説明させていただきます。これは私が考えた例ではないことをあらかじめことわっておきます。まず例を2つあげます。</p> <p>①LD GR1,#0102 ②LAD GR1,#0102</p> <p>(シャープがついているとき、これは16進数で表記されていることを意味します) #0102=(0102)h ということですね。</p> <p>①の例ですが、LD 命令の場合には、オペランドの#0102 を主記憶に対するアドレスとして、主記憶のアドレス#0102 番地にアクセスし、その番地に格納されている値を読み込んで、汎用レジスタ GR1 に設定します。一方、②の例ですが、LAD 命令の場合には、オペランドで指定された#0102 をそのまま汎用レジスタに設定します。</p> <p>つまり、LAD 命令では主記憶にアクセスしないという点がポイントです。少々小難しいことを書いてしまいましたが、理解していただけたら幸いです。</p>
学習者 B	2007/06/13 01:15:38	ありがとうございます。
学習者 C	2007/06/16 23:14:20	<p>横から突然申し訳ありませんが、私も LD と LAD の違いを説明させてください。</p> <p>LD は指定したアドレス (番地) に入っている数字や文字を得ます。</p> <p>LAD は指定したアドレスの番地を得ます。</p> <p>例えば、ORG (5 番地) に 'あ' が入っていたとします。</p> <p>LD GR1,ORG ならば、GR1 は 'あ' を得ます。</p> <p>LAD GR1,ORG ならば、GR1 は 5 を得ます。</p> <p>こんな感じではないでしょうか？何か意見や訂正があれば教えて下さい。</p>
学習者 D	2007/06/17 22:43:50	<p>横の横ですいません。</p> <p>学習者 C さん、とても分かりやすい説明で助かりました。</p>
学習者 A	2007/06/17 22:51:10	学習者 C さんへ>全くその通りだと思います。
学習者 E	2007/06/18 07:17:38	<p>ありがとうございます！</p> <p>助かりました</p>
学習者 F	2007/06/22 16:43:16	<p>たしかにこの方がわかりやすいと思います。</p> <p>LD, LAD の使い分けは慣れですね(>_<)</p>

3.4.3 システム利用実験から得られるデータ

システム利用時に、各利用者の行動は特定のタイミングで行動履歴として記録される。行動履歴には「行動した利用者」、「行動時刻」、「行動内容」がセットとなって記録され、このデータを利用者別に集計することで行動回数を得ることができる。

履歴として記録される行動には具体的に、「ログイン」、「問題一覧閲覧」、「問題作成」、「問題解答開始」、「問題解答終了」、「評価登録」、「評価一覧閲覧」、「解答履歴閲覧」、「問題修正」、「掲示板投稿」、「掲示板スレッド一覧閲覧」、「掲示板閲覧」の12種類ある。行動履歴とは別に、Concerto III の評価登録によって蓄積された評価の値も利用できる。

各利用者のシステム利用前のプレテストの成績、およびシステム利用後のポストテストの成績もデータとして利用できる。

3.5 問題作成数および作成した問題の正確性と成績との関連

本節では、ConcertoIIIの2008年度の適用実験結果とConcertoIIIを適用した2008年度の間接テストと期末テスト結果をもとに、本研究で開発した作問学習支援システムの適用効果を分析する。分析では、問題作成数を作成された問題に対して学習者（その問題を評価した学習者）が行った問題の正確性評価を用いた分析を行った。さらに、システムが学習者に与える影響の調査として、システム利用アンケートを用いた分析を行った。なお、本節で示す統計技法を用いた検定は全てSPSS for Windows 13.0Jで行った。

3.5.1 問題作成数を用いた試験結果の分析

問題を作成することについて、Yuらは「多肢選択問題を生成する作業を行わせると、生徒は問題文と正しい答えと（正解以外の）選択肢を作成する必要がある。そのプロセス中に生徒は、最初にその教材のどの部分が重要でテストをする価値があるかを考えなくてはならない。そして生徒は、戦略的に問題を言い表し、正しい答えを考える必要がある。また、生徒は問題文の適正さ、答えの正しさ、（正解以外の）選択肢の良さを厳密に検査する必要がある。これらの活動にはリハーサル、推敲のような認知プロセスを含み、情報処理理論の見地から認知発達のために有益である。」と述べている[Yu, 2005]。

本研究では、多肢選択問題だけでなく、記述式問題も作成できるようにしている。記述式問題の作成においても、問題文の適正さ、答えの正しさを厳密に検査する必要があるため、多肢選択問題の作成と同等の効果が得られると考えられる。問題を作成することが認知発達のために有益である[Yu, 2005]とのことから、問題を多く作成している学習者ほど認

知発達が著しいと考えられる。その結果、問題を多く作成している学習者ほど、システム利用後の期末テストで良い成績を取るのではないかと考えられる。

そこで、この仮説を検証するため、ConcertoIIIを適用した2008年度の間接テストと期末テスト結果を用いて、問題を多く作っている学習者とそうでない学習者に分けて、中間テストや期末テストの成績の平均値を比較した。

平均値の比較はt検定を利用する。本節でのt検定では、問題を多く作っている学習者群とそうでない学習者群、つまり2つの群がそれぞれ独立であるとして、それぞれの群の母集団の平均値が等しいかどうかの検定を行う。母集団が満たしている条件によって、実施するt検定が異なるため、t検定を行うために次の2つの検定を実施する。

- 各標本が正規分布に従うかどうかの検定：コルモゴロフ-スミルノフ検定、またはShapiro-Wilk検定
- 標本の分散が等しいかどうかの検定：F検定、またはLevene検定、またはバートレット検定

SPSS for Windows 13.0Jでは、標本（実データ）を入力するだけで、t検定をおこなうための2つの検定、およびそれに応じて実施したt検定の結果を同時に出力するため、本研究では、その出力した結果をもとにしている。以下の記述では、t検定の結果のみを示す。

表3-4にConcertoIIIにおける問題作成数の多少で学習者を分割したときの各群における中間テスト成績の結果を示し、図3-6に成績の分布を示す。問題作成数の多少は、システムを利用した全学習者の問題作成数の平均3.01問を境として分割した。この2群の中間テスト成績の平均点について、危険率5%のt検定（両側検定）を行った結果 $p=0.796$ となり、有意差は認められなかった。

同様の分析を、期末テスト成績を用いて行った。結果を表3-5に示し、成績の分布を図3-7に示す。2群の期末テスト成績の平均点について、危険率5%のt検定（両側検定）を行った結果 $p=0.038$ となり、有意差は認められた。

この結果、中間テストにおいて2群の成績は同等であることが認められ、期末テストにおいて問題を多く作成していた群の成績がそうでない群の成績よりも高いことが認められた。したがって、問題を多く作成している学習者はそうでない学習者よりも中間テストの成績から期末テストへの成績の伸びが大きかったと考えることができる。

表 3-4 ConcertoIIIの問題作成数の多少による中間テスト結果の分析

	問題作成数 が多い	問題作成数 が少ない	p 値
人数	16	31	
平均点 (標準偏差)	53.13 (30.923)	55.16 (22.490)	0.796

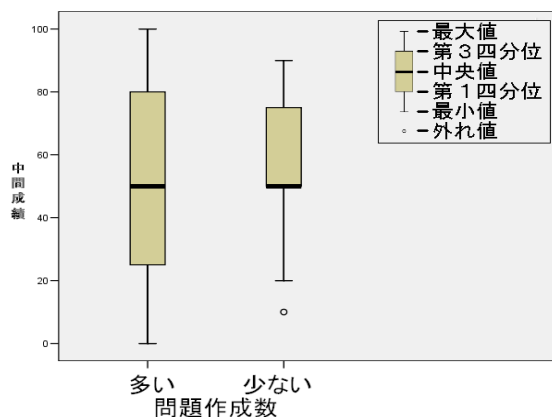


図 3-6 ConcertoⅢの問題作成数の多少による中間テスト成績の分布

表 3-5 ConcertoⅢの問題作成数の多少による期末テスト結果の分析

	問題作成数 が多い	問題作成数 が少ない	p 値
人数	23	63	
平均点 (標準偏差)	60.39 (12.551)	51.48 (18.868)	0.038

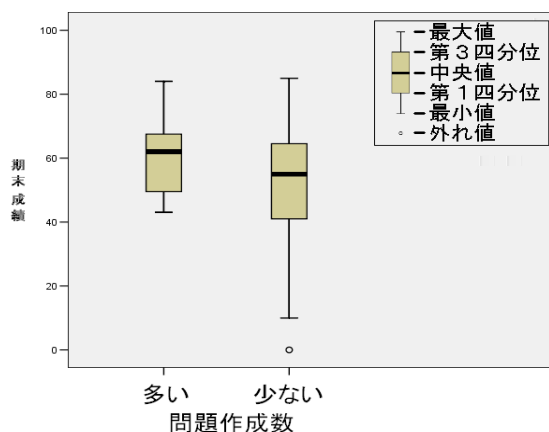


図 3-7 ConcertoⅢの問題作成数の多少による期末テスト成績の分布

3.5.2 問題の正確性評価データを用いた試験結果の分析

問題の正確性評価データとは 3.3.2 項(2)で述べた信頼性評価である。問題を作成する際、問題文や正答、解説に誤りがないかをチェックするため、学習者は問題文や正答、解説を何度も読み返し、必要に応じて教科書等の参考資料を参照する必要がある。これを繰り返している学習者は、その学習者が作成した問題に対する他学習者の問題の正確性評価が高くなる傾向があると考えられる。本研究における問題の正確性評価は問題文、解答、

解説の適正さ、正確さ（誤字、脱字、曖昧な表現がない）を評価基準として示している。これらの活動が認知発達に有益であるといわれているため[Yu, 2005]、問題の正確性評価が高い学習者ほど認知発達が著しいと考えられ、システム利用後の期末テストで良い成績を取るのではないかと考えられる。

この仮説を検証するため、中間テストや期末テストの成績を用いて、正確性の高い問題を多く作っている学習者とそうでない学習者に分けて、中間テストや期末テストの成績の平均値を比較した。

(1) 問題の正確性評価データとその妥当性

本研究では、問題の評価は問題に解答した学習者が行うため、問題の正確性評価の妥当性を検証する必要があると考えた。

そこで、ConcertoIIIに登録された273問のうち、評価が1件以上あった265問を対象として、その問題を評価した「学習者による問題の正確性評価の平均値」と「教授者による問題の正確性評価の値」の相関を分析し、学習者と教授者の問題の正確性評価が近似しているかどうかを調査した。

図3-8に学習者による問題の正確性評価と教授者による問題の正確性評価の散布図を示す。この2標本の相関係数は0.324（無相関検定の有意確率 $p=0.000$ ）であり、学習者による問題の正確性評価と教授者による問題の正確性評価には弱い正の相関があると判断できる。

学習者が評価した問題の正確性評価に妥当性があるかどうかを判定するにはさらなる調査（たとえば、評価した学習者の学習内容の理解度を測定する）が必要であるが、本研究では、学習者による問題の正確性評価と教授者による問題の正確性評価はおおよそ近似しているとみて、学習者の評価を用いて分析を行うこととした。

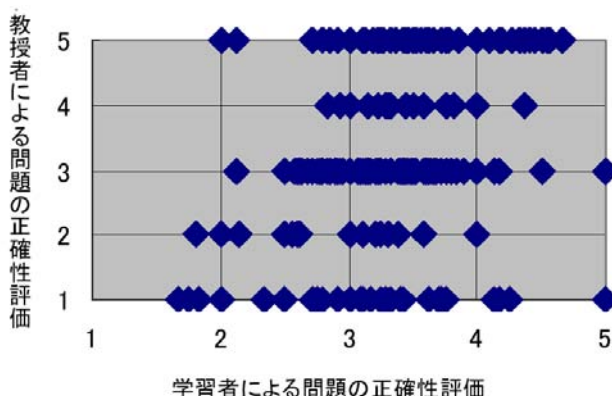


図3-8 学習者による問題の正確性評価と教授者による問題の正確性評価の散布図

(2) 試験結果の分析

表3-6にConcertoIIIにおける問題の正確性評価の高低で学習者を分割したときの各群における中間テスト成績を示し、図3-9に成績の分布を示す。問題の正確性評価平均の高低は、

表 3-6 ConcertoⅢの問題の正確性評価平均の高低による中間テスト結果の分析

	問題の正確性 平均が高い	問題の正確性 平均が低い	p 値
人数	25	20	
平均点 (標準偏差)	52.00 (24.324)	56.00 (27.796)	0.610

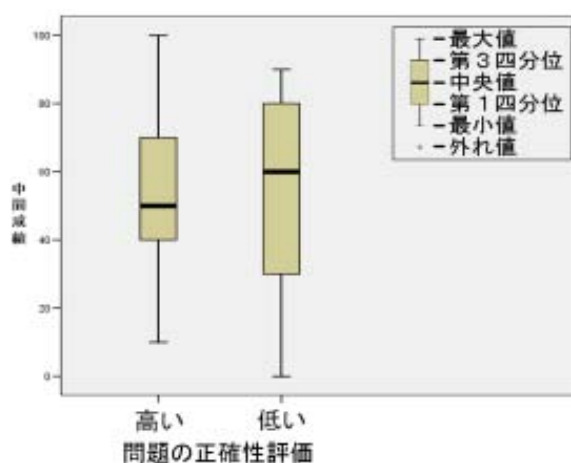


図 3-9 ConcertoⅢの問題の正確性評価平均の高低による中間テスト成績の分布

表 3-7 ConcertoⅢの問題の正確性評価平均の高低による期末テスト結果の分析

	問題の正確性 平均が高い	問題の正確性 平均が低い	p 値
人数	43	38	
平均点 (標準偏差)	59.72 (14.498)	49.82 (18.589)	0.008

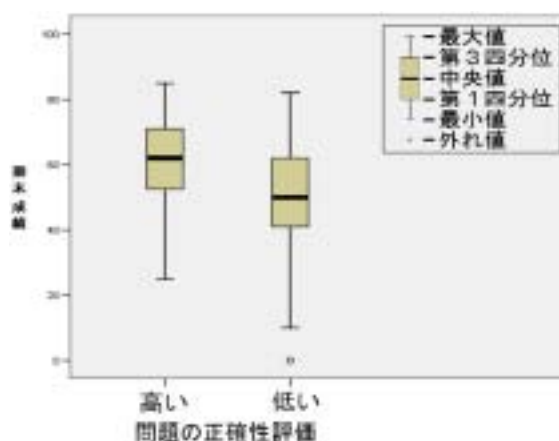


図 3-10 ConcertoⅢの問題の正確性評価平均の高低による期末テスト成績の分布

ConcertoⅢで問題を作成し、作成された問題に対する評価を1件以上受けた全学習者の問題の正確性評価の平均3.29を境として分割した。この2群の中間テスト成績の平均点について、危険率5%のt検定（両側検定）を行った結果 $p=0.610$ となり、有意差は認められなかった。

同様の分析を期末テスト成績の平均点を用いて行った。結果を表3-7に示し、成績の分布を図3-10に示す。この2群の期末テスト成績の平均点について、危険率5%のt検定（両側検定）を行った結果 $p=0.008$ となり、有意差は認められた。

この結果、中間テストにおいて2群の成績は同等であることが認められ、期末テストにおいて正確性の高い問題を作成していた群の成績がそうでない群の成績よりも高いことが認められた。したがって、正確性の高い問題を作成する学習者はそうでない学習者よりも中間テストから期末テストへの成績の伸びが大きかったと考えることができる。

3.5.3 システム利用アンケート

ConcertoⅢの利用アンケートにおいて、被験者に「ConcertoⅢの利用前と利用後で何か変化したことはあるか?(ただし、計算機システム概論の内容が分かった以外の記入とする)」という質問への回答を依頼した。これに対し、被験者から以下の回答を得た（質問に的確に回答していない内容を除くすべての回答を列挙）。

- 計算機システム概論という分野に、より積極的に取り組めるようになった
- 学習に対する意識が上がった
- モチベーションは上がった。実力もついていることをいりたい
- 自分が問題を作る上での主旨と他の人とのそれとの違いがあつて、それがテスト勉強をする上で役立った。（“こんな事をやらなくては”と）
- テストに向けての意識向上。他の人と議論する中で様々な解答方法があることが分かった
- 勉強意欲が増した。もっと早くやればよかったと思った
- 勉強方法が増えた
- 情報の勉強をより身近に感じる事ができた
- 分からなかったところが分かって勉強する材料を与えてくれてよかったです
- 意識が変わった。また勉強のやる気が上がった
- 計算機システム概論が好きになった
- 問題を解いているうちに楽しくなり、もっとやりたいと思った。試験前の勉強に最適
- もっと解きたいという気持ちになった
- ちょっとの時間でもパソコンを開いて勉強するようになった
- 特に変化はなかった。ConcertoⅡの方は良く使っていたと思う。あまり機能を追

加しすぎるのは・・・

- 私だけでなく、他の人もたくさん勉強しているのが見え、自分も負けるものかと思ひ、やる気になりました
- もっとはやくから活用すればよかった。テスト前あせっているときにはじめても100%活用できない
- モチベーション。表現力
- 利用前は、計算機システム概論は大嫌いと思っていたけど、利用することで理解が深まり問題を解くのがどんどん楽しくなっていた。実際アセンブラの問題を解くのにはまりすぎた。もっと早く ConcertoIIIのおもしろさに気づいていたらよかったなあと思う
- 普通に教科書を見て勉強するよりも学力がついたと思う

システム利用前とシステム利用後の変化として、学習に対する意識が変わった（学習をする意欲がわいた）と回答している学習者が多く、もっと早くから ConcertoIIIを利用すれば良かったと後悔している学習者も少なくなかった。これらからシステムが学習者の学習に対する意識に刺激を与えているのではないかと考えることができる。

3.6 問題作成数，問題の正確性を用いた分析に対する考察

3.5.1 項の結果から、問題を多く作成することが学習効果につながると考えられる。3.5.2 項の結果から、正確性の高い問題を多く作成することが学習効果につながると考えられる。本節では、これらの結果について考察する。

(1) 環境の違いによる学習効果への影響

本章では、分散非同期環境においても作問学習による学習効果が得られているかという観点で分析を行った。分散非同期環境における作問学習と、対面環境における作問学習では、環境の違いが学習効果に及ぼしうる影響があると考えられる。例えば、Yuらは「Webベース上でのやりとりになることで相手の状況が確認できない等の心理的負荷が発生し、それによって本来の力が発揮できなくなる」と述べている[Yu, 2008]。また、作問してから、解答や評価するまでにタイムラグが発生することによる影響も考えられる。

ConcertoIIIでは、Yuらの研究[Yu, 2008]で報告されていることと同様、学習者に心理的負荷を与えないように、ユーザ名を本名ではなくハンドルネームを利用している。また、作問してから解答や評価するまでにタイムラグがあった場合を想定して、アウェアネス機能（例えば、作成した問題に設置されている掲示板に新規投稿があったことを作問者に知らせる）を提供している[平井, 2008]。

しかし、これらが学習効果に影響を及ぼしているかどうかは、今回の結果からだけでは

判断できない。これについては今後調査していく必要がある。

(2) 問題の正確性評価の指標

3.5.2 項で述べた問題の正確性による成績の伸びの調査について、本研究では、問題の正確性を「問題文、解答、解説の適正さ、正確さ（誤字、脱字、曖昧な表現がない）がないこと」と定義したため、これを作成された問題の評価観点としている信頼性評価を用いて分析を行った。

しかし、作問をする上では信頼性評価以外の項目（例えば、本稿で挙げた新規性評価、難易度評価）も重要であり、これが学習効果に影響を及ぼしている可能性がある。これについては今後調査していく必要がある。

(3) 学習意欲の影響

成績を利用した分析において、中間テストでは 2 群とも同等の成績であったのが、期末テストでは、問題作成数の多い学習者はそうでない学習者よりも平均点が高かった。同様に、正確性の高い問題を多く作成した学習者はそうでない学習者よりも期末テストの平均点が高かった。

これらから、問題を多く作成することや正確性の高い問題を多く作成することが、学習効果につながる可能性があると考えられる。しかし、例えば「計算機システム概論」に積極的に取り組んでいる者の期末テストの成績が良いことを示している可能性もある。また、問題の正確性についても、積極的に取り組んでいる者は意欲が高いため、結果として、正確性が高いものになっている可能性もある。これについては今後調査していく必要がある。

3.7 相関ルールマイニングを利用した分析

相関ルールは、トランザクションデータベースに頻出するアイテム間の何らかの組み合わせのルールを指す[Agraval, 1994][浜野, 2007]。例えば事象 A が発生したときに事象 B が発生するという相関ルールを「 $A \Rightarrow B$ 」と表わす。ルールの左辺 A を条件部、右辺 B を結論部と呼ぶ。

データベース内に無数に存在するルールの中から有意義な相関ルールを検出するには、何らかの評価指標が必要である。そのため信頼度、支持度といった指標が一般的に利用されている。

相関ルールの信頼度 **confidence** は、事象 A が発生した前提のもと事象 B が発生する確率、つまり条件付き確率で定義される。信頼度が 1 に近いほど前提と結論の結びつきが強いといえる。

$$\text{confidence} (A \Rightarrow B) = p (B|A)$$

支持度 **support** は事象 A と事象 B がトランザクション全体の中に占める確率，つまり共起確率で定義される。

$$\text{support } (A \Rightarrow B) = p(A \cap B)$$

支持度が高ければそのルールが優れているということとはできないが，支持度の低すぎるルールはそのデータベース内でほとんど出現しておらず，信頼度が高かったとしても説得力に欠ける．このように相関ルールの評価には信頼度と支持度の双方を考慮する必要がある．実際のルール抽出においては，閾値を設定し，一定の信頼度と支持度の値を超えるルールのみを抽出するといった手法がとられる．

3.7.1 相関ルールの抽出

本項では，相関ルールの抽出手法についてその手順に沿って説明する．相関ルールの抽出手法は浜野らの論文[浜野, 2007]及び Hamalainen らの論文[Hamalainen, 2006]で行われている手法に従った．

(1) データの 2 値化

まず各利用者に対する行動回数などの集計を行った．集計結果の一部を表 3-8 に示す．この表では，次のデータについて学習者ごとに集計した．

- システムから得られるデータ：ログイン数，問題作成数，問題解答数，問題評価数，掲示板投稿数，問題の正確性 (=信頼性評価の平均値)
- 試験成績：プレテスト点数，ポストテスト点数

集計された数値をそのまま使うと抽出されるルール数が少なくなるため，数値として与えられる項目のデータの 2 値化を行い，データを 2 つのグループに分類する．分類結果の一部を表 3-9 に示す．問題作成数，問題解答数等の項目では，2 値化したときに両グループの学習者数ができるだけ同じになるように「平均値以上 (H=High)」と「平均値未満 (L=Low)」の 2 グループに分けた．成績を見る指標は，成績が向上したかあるいは低下したかを見るため，ポストテストとプレテストの偏差値の差をとり，その正負で 2 グループに分類した．差分が正のグループは成績が向上し，そうでないグループは成績が低下したか変化がなかったと考えた．

表 3-8，表 3-9 のような表を作成する作業について，2008 年度，2009 年度のそれぞれのデータを用いて行い，2 値化されたものについて両年度のデータを結合し，119 人分のデータを作成した．

(2) ルールの抽出

表 3-8 のように，データの 2 値化を行った後，支持度と信頼度の算出および相関ルールの抽出を行う．これらの処理は統計処理ソフトウェア R 2.9.1[R Project]に相関ルールマイ

ニング用の `arules` パッケージ[arules]を加えて利用し、`apriori` アルゴリズム[Agraval, 1994]による相関ルール抽出を行った。

(3) 結論の絞込み

本研究の目的は学習効果の向上につながる要因を分析することであるため、抽出されたルールに対して結論部が「成績=向上」であるもののみを抽出した。また同時に、結論部が「成績=低下」であるものについても抽出した。この後、学習・教育的に意味のあるルールに絞るための条件を設定した。

まず、対象とする学習者を絞る。前述のとおり、成績に関する 2 値化は、プレ、ポストテスト間の偏差値差分について単純に向上か低下かに分けたが、このようにすると、成績がわずかに向上、低下したものの、成績がほとんど変化していない学習者も混在している。そこで、(1)で得られた 119 人に対して (ポストテストの偏差値) - (プレテストの偏差値) の大きい順に並べ替え、その上位 33% (39 人) と下位 33% (39 人) のみを対象とした。

表 3-8 行動履歴の集計

利用者	問題作成 (回)	問題解答 (回)	…	プレテスト (点)	ポストテスト (点)
U1	4	30	…	6	62
U2	6	218	…	5	73
…	…	…	…	…	…
Un	3	65	…	7	48
平均	2.93	41	…	-	-

表 3-9 集計結果の 2 値化

利用者	問題作成数		問題解答数		…	成績	
	H	L	H	L		向上	低下
U1	1	0	0	1	…	0	1
U2	1	0	1	0	…	1	0
…	…	…	…	…	…	…	…
Un	0	1	1	0	…	0	1

(1…グループに属する, 0…グループに属さない)

次に最低支持度および最低信頼度の設定を行う。信頼度について、この値が低すぎるとルールを満たす学習者が少なくなるため、ルール抽出の条件として最低支持度が 0.15 にな

るよう設定した。これにより、分析対象者全体のうち少なくとも12人以上 ($78 \times 0.15 = 11.8$) が満たしているルールのみ抽出できる。また、最低信頼度について、この値が低いと条件部に対して「成績=向上」と「成績=低下」の確率が近づいてしまうため、ルール抽出の条件として最低支持度を0.55になるよう設定した。これにより、「成績=向上」と「成績=低下」の確率に10%以上の差が開くように抽出できる。

3.7.2 抽出結果

前項の手順に従って成績の向上と関連があるルールを抽出した結果、成績の向上と関連があるルールは7件、成績の低下と関連があるルールは12件抽出された。抽出されたルールを表3-10、表3-11に示す。両表ともに抽出されたルールについて信頼度の高い順に、信頼度が同じ場合は支持度の高い順に信頼度順位をつけている。

表3-10、表3-11では、各行が1つのルールを表しており、例えば、表3-10の信頼度順位1位のルールは、「(問題解答数=H) \cap (掲示板投稿数=L) \Rightarrow (成績=向上)」という形式のルールが抽出され、その支持度は0.153、信頼度は0.666だったことを示している。言い換えれば、問題解答数が平均値より高く、掲示板投稿数が平均値より低く、成績が向上した学習者は、全学習者のうち約15%いて、条件部を満たす学習者のみ対象とすると、そのうち成績が向上した学習者は約67%いたというルールである。

表 3-10 成績の向上と関連するルール

順位	左辺 (条件部)	支持度	信頼度
1	問題解答数=H, 掲示板投稿数=L	0.153	0.666
2	問題解答数=H	0.192	0.652
3	ログイン数=H, 掲示板投稿数=L	0.153	0.631
4	問題評価数=H	0.179	0.608
5	問題解答数=H, 問題評価数=H	0.153	0.600
6	掲示板投稿数=L, 問題の正確性=H	0.346	0.562
7	ログイン数=L, 問題の正確性=H	0.230	0.562

ルールは左辺のみ示す。右辺は全て「成績=向上」である。

表 3-11 成績の低下と関連するルール

順位	左辺 (条件部)	支持度	信頼度
1	ログイン数=L, 掲示板投稿数=L, 問題の正確性=L	0.153	0.631
2	ログイン数=L, 問題の正確性=L	0.153	0.600
3	問題解答数=L, 問題評価数=L	0.384	0.576
4	問題作成数=L, 問題の正確性=L	0.153	0.571
4	掲示板投稿数=L, 問題の正確性=L	0.153	0.571
6	問題解答数=L, 問題評価数=L, 問題の正確性=H	0.269	0.567
7	問題解答数=L	0.397	0.563
8	問題作成数=L, 問題解答数=L, 問題評価数=L	0.307	0.558
9	問題評価数=L, 問題の正確性=H	0.269	0.552
10	問題解答数=L, 問題の正確性=H	0.282	0.550
10	ログイン数=L, 問題作成数=L, 問題評価数=L	0.282	0.550
10	ログイン数=L, 問題作成数=L, 問題解答数=L	0.282	0.550

ルールは左辺のみ示す。右辺は全て「成績=低下」である。

3.8 相関ルールマイニングを用いた分析に対する考察

3.8.1 既存研究との比較

本項では、作問学習支援システムに関する定量的な分析を行った既存研究[高木, 2007][平井, 2010][Hirai, 2009]で報告されたことと本研究の結果を比較する。平井らの研究[平井, 2010][Hirai, 2009]では「計算機システム概論」、高木らの研究[高木, 2007]では「情報処理」という講義内で作問学習活動が行われている。これらの既存研究は、大学生を対象とした Web ベースの作問学習支援システムに関して分析を行っており、それらが試験結果へどう影響するかについて述べている。

(1) 問題を作成することの効果

文献[高木, 2007]では、システム利用アンケート結果から問題を作成することの効果があったと述べているが、問題作成数の多少による試験結果への影響はないと述べている。文献[平井, 2010]では、問題を多く作成する学習者はそうでない学習者と比べて、試験結果に

良い影響を与えると述べている。

これらについて、表 3-10 あるいは表 3-11 に「問題作成数=H」という要因が含まれるルールが抽出されていないことから、問題作成数の多少が試験結果に影響するかどうかは、抽出結果からだけでは判断することができない。しかし、表 3-11 のみに「問題作成数=L」という要因が含まれるルールがあることから、問題作成数が平均値より少ないことと成績低下が関連すると推測することは可能である。

(2) 正確性の高い問題を作成することの効果

文献[平井, 2010]では、正確性の高い問題を作成する学習者はそうでない学習者と比べて、成績向上に良い効果があると報告している。

これについて、表 3-10 に「問題の正確性=H」が、表 3-11 に「問題の正確性=H」及び「問題の正確性=L」という要因が含まれるルールがあることから正確性の高い問題を作成することが試験結果に影響するかどうかは、抽出結果からだけでは判断することができない。しかし、表 3-11 のみに「問題の正確性=L」という要因が含まれるルールがあることから、平均的に正確性の低い問題を作成することが成績低下に関連すると推測することは可能である。

(3) 作成された問題を多く評価することの効果

文献[Hirai, 2009]では、作成された問題を多く評価する学習者はそうでない学習者と比べて、成績向上に良い効果があると報告している。

これについて、表 3-10 のみに「問題評価数=H」が、表 3-11 のみに「問題評価数=L」という要因が含まれるルールがあることから、問題評価数が平均値より高いことと成績向上が関連すると推測することは可能である。

3.8.2 相関ルールマイニングを行うことの利点

前項で既存研究との比較を行ったが、どの項目についても推測の域をでることはしなかった。例えば問題を作成することについて、表 3-10 に「問題作成数=H ⇒ 成績=向上」というルールが、表 3-11 に「問題作成数=L ⇒ 成績=低下」というルールが抽出されていれば、問題作成数が多いことが成績向上に良い効果があると言うことの妥当性は上がる。さらに妥当性を上げるには、問題作成数のみに注目して、t 検定等の統計処理を行えばよい。このため 3.8.1 項では推論に留めた。

しかし、相関ルールマイニングを行うことにこれまでは発見できなかったことが分かることもある。例えば、表 3-10 の信頼度順位 1 位と 2 位のルールについて見ると、「問題解答数=H」という単独の要因で構成されるルールの信頼度は 0.652 であるが、「掲示板投稿数=L」という要因を組み合わせると信頼度が向上している。この例は表 3-11 も含めていくつ

かある。また、「問題解答数=H ⇒ 成績=向上」というルールの信頼度は「問題評価数=H ⇒ 成績=向上」というルールの信頼度より高いことから、問題解答数が平均より高いことが、問題評価数が平均より高いことよりも、成績向上と大きく関係していると解釈することが可能である。

このようにルールの信頼度同士で比較することや、複数要因を組み合わせることで、「成績=向上」あるいは「成績=低下」とより関連が強いルールを発見することができる。このような発見はt検定等の統計処理で発見することは難しく、相関ルールマイニングを用いることの利点であると考えられる。

3.8.3 相関ルールマイニングの限界

前項まで説明してきたように、相関ルールマイニングは支持度と信頼度の閾値を設定し、それ以上の値を持つルールだけを抽出することができる手法である。この手法では、抽出されるルール数が多くならないように閾値を設定する。本研究では、支持度を0.15以上、信頼度は0.55以上を満たすルールを抽出した。設定しただけで抽出されるルール数が変わり、それに伴って抽出結果の解釈も変わる。閾値を大きくしてルールを減らすとその内容は既知のことばかりとなり、解析自体が無意味になる[植野, 2007]と指摘されている。

また本研究では学習者を2つのグループに分けるときに、2つに分けたときの学習者数ができるだけ同じ人数になるように平均値をもとにして分けた。当然ながら、2つに分ける基準を平均値以外にすれば、各学習者の属するグループ(H or L)が変わり、抽出されるルールの支持度、信頼度が変わることもある。閾値の設定やグループ分けをする基準を変更すると、抽出されるルールが変わるため、これらを設定する際には十分な検討が必要である。

3.8.4 作問学習支援システムを用いることの効果

これまでの考察を踏まえ、本項では抽出された相関ルールから分かる作問学習支援システムを利用することの効果について考察する。

(1) 掲示板投稿活動による効果

表 3-10, 表 3-11 の1位ともに「掲示板投稿数=L」が含まれるルールが抽出されていることから、掲示板投稿数が学習者全体の平均値より少ないことが、成績の向上、低下どちらにも影響したと解釈できる。

最低信頼度を設定したために表 3-10, 表 3-11 には「掲示板投稿数=L」という単独の要因が含まれるルールが抽出されていないが、「掲示板投稿数=L ⇒ 成績=向上」の支持度は0.462, 信頼度は0.521であった。3.7節で述べた定義に従うと「掲示板投稿数=L ⇒ 成績

=低下」の信頼度は 0.479 (=1-0.521) である。つまり、「掲示板投稿数=L」を満たしている学習者が多く、かつ成績が向上するか低下するかの確率がほぼ同じであることから、表 3-10, 表 3-11 とともに「掲示板投稿数=L」という要因が含まれたルールが多く抽出されたものと考えられる。

よって、掲示板投稿活動による成績への影響は今回の分析結果からは述べることはできない。

(2) システムの積極的利用による効果

表 3-10 では「ログイン数=H」、「問題解答数=H」、「問題評価数=H」という要因が含まれるルールがあるように学習者全体の平均よりも多くの活動を行っている学習者は成績向上との関連が強く、表 3-11 では「問題解答数=L」のような要因が含まれるルールが見られることから、システムを積極的に利用している学習者は成績向上と関係していると考えられることができる。これはシステムを積極的に利用していない学習者よりも、当該学習内容に対して多くの時間関わっているため、成績向上と関係するものと考えられる。

(3) 問題解答活動による効果

表 3-10 の 2 位に「問題解答数=H ⇒ 成績=向上」、表 3-11 の 7 位に「問題解答数=L ⇒ 成績=低下」というルールがあることから、問題解答数の多少が成績の向上、低下に関係していると解釈できる。問題を解くことに学習効果があることは、当然とされているが、この結果はこの事実を改めて示しているといえる。

(4) 作成する問題の正確性による効果

作問学習研究に関して注目すべき点は、問題の正確性による影響である。これらについて 3.8.1 項で推論を述べたが、表 3-10 と表 3-11 ではこれに関する要因が含まれるルールが多く抽出されていることから、成績の向上、低下に関係する要因であると考えられる。特に「問題の正確性=L」については、(1)で述べた「掲示板投稿数=L」の場合とは違い、「問題の正確性=L」を含むルールが表 3-11 のみにあることから、「問題の正確性=L」が成績低下と大きく関係していると考えられる。

問題の正確性を高めるには、問題を作成する際に「問題文に誤字・脱字が少なく、解答・解説が間違っていない」ようにする必要がある。これはただ単に問題を作成する以上の作業が必要である。この作業には、例えば、誤字・脱字を少なくするために問題文を注意深く作成するということや、解答・解説が間違っていないかどうかを調べるために問題を一度解いてみることや他学習者の視点になって問題を客観的に評価するということも含まれると考えられる。また、Yu らの研究[Yu, 2005]では、客観的に問題を評価することで、既存の知識に新しい知識を追加して知識を再構築したり、誤解を正したりできるであろうと述べている。

つまり、問題の正確性を高めるためには、問題を作成するだけでなく、他学習者の立場になって問題に解答したり、問題を評価したりする作業も含まれる。このことが成績に影響しているのではないかと考えられる。

3.9 まとめ

本研究では、学習者による作問に基づく学習支援システム:ConcertoIIIを開発し、これを大学の講義における分散非同期環境での学習環境として適用した。

本章ではその適用実験結果を報告し、分散非同期環境においても作問学習による学習効果が得られるかどうかという観点から分析を行った。特に、分散非同期環境において、作成する問題文・解答・解説の内容に誤りがないという問題の正確性を用いた学習効果の分析を行った。その結果、作問を多く行っている学習者はそうでない学習者よりも期末テストへの成績の伸びが大きく、正確性の高い問題を作成する学習者はそうでない学習者よりも期末テストへの成績の伸びが大きかったことが分かった。また、システムの導入により、システムが学習者に対して学習のきっかけを与えていたことが分かった。これらから、分散非同期環境における作問学習でも、学習による効果が得られるのではないかと考えることができた。

また本研究では、作問学習支援システムにおける学習者の活動を、相関ルールマイニングを用いて分析した。これにより、ポストテストの成績がプレテストの成績より向上あるいは低下した学習者について、システム内における作問学習活動と成績の関係を調査した。抽出したルールを解釈した結果、次のことが明らかになった。

- システムを積極的に利用すること、学習者が作成した問題に多く解答すること、学習者が作成する問題の正確性が高いことが、成績向上に関係している。(3.8.4 項(2)(3)(4))
- 問題に解答することが成績向上に大きく関係している(3.8.2 項, 3.8.4 項(3))
- 正確性の低い問題を作成することが成績低下に大きく関係している(3.8.4 項(4))

特に2点目、3点目は既存研究では報告されておらず、今後、作問学習を支援する環境を開発することにおいて重要な知見となろう。

第4章 対面環境における協調プログラミング学習の分析

本章では、対面環境における協調プログラミング学習の分析として、対面同期環境で実施したペアプログラミング学習活動を分析した結果を報告する。

ペアプログラミング学習を実施することにした理由は学習者同士のコミュニケーションがより密に行われるからである。2.2.1項で述べたように、グループサイズが2人であるとコミュニケーションが密に行われることが指摘されている[中野, 2003]。また、協調学習では、最終的な成果を出すまでに至るプロセスの質が重要となる[稲葉, 1999]という指摘もある。これらの指摘を受け、本研究では、ペアプログラミング学習を実施し、ペアプログラミングの成功事例と失敗事例を比較分析した。

4.1 研究概要

プログラミング教育では、プログラム言語の文法やプログラム書法を理解する能力とアルゴリズムを組み立てる能力が要求される。プログラム言語の文法や簡単な例題を理解することができても、実際にプログラムを作成するとなったとき、アルゴリズムを組み立てることができなければ、プログラムを組むことができない[新開, 2009]等の問題が生じる。問題の解決を支援する実践も含め、プログラミング教育では、これまでに多種多様な実践が行われており、情報処理学会誌でも2010年10月号からプログラミング教育に関する連載が行われた[大西, 2010]等、その教育方法や学習方法に関して広く注目されている。

プログラミングを行う方法の1つとして、プログラミングを2人1組のペアになって行うペアプログラミングがある。ペアプログラミングはBeckら[Beck, 1999]によって提唱されたエクストリーム・プログラミング(XP)と呼ばれるソフトウェア開発手法の1つのコンポーネントである。ペアプログラミングでは、1人がDriverと呼ばれる役割として実際のコードを書き、もう1人はNavigatorと呼ばれる役割として、Driverをチェックしながらナビゲートし、この役割を随時交代しながら作業を進める。これによって、問題解決に要する時間を短縮させる、常にコードレビューを行うことができる等の効果が得られる。ペアプログラミングによるプログラム作成は協調作業であり、また、プログラミングに関する専門知識の共有および拡張という点でも有効という指摘があり[Wray, 2010]、知識の獲得、拡張に有効なコラボレーション手法であると考えられる。

このペアプログラミングを用いたプログラミング学習が行われている。特に、プログラミング導入教育におけるペアプログラミングの利用では、個人でプログラミングをするときと比較した場合に、プログラムの質が向上する、課題解決までの時間が短縮する等、そ

の有効性が報告されている[McDowell, 2002][Nagappan, 2003][Hanks, 2004][Rountree, 2005]. しかしながら, いくつかの実践において一部のペアでは与えられたプログラム作成課題の解決がうまくいっていないという問題は発生している. ペアの組み合わせ等によってペアプログラミングを行ったときに得られる効果が異なることがあると考えられるが, プログラミング導入教育では, プログラミングの基本事項を学習するために, 得られる効果に違いがでてしまうと, 後々の学習に悪影響を及ぼす可能性がある. 特に, 与えられたプログラム作成課題の解決がうまくいかないと, 学習に対するモチベーションの低下等が懸念される. そこで, プログラミング導入教育におけるペアプログラミング学習において, プログラム作成課題の解決がうまくいっていないペアを支援することを考えた.

本研究では, プログラミング導入教育においてペアプログラミング学習を実施し, ペアプログラミングの成功事例と失敗事例を比較分析した. 分析では解決中に会話を行っているペアの会話に着目し, 失敗事例の方が発話が長いこと, 説明の繰り返しが多いこと, 一方的な発話が多いことが分かった. この知見は, ペアプログラミングにおいて協調作業がうまく進んでいるかどうかを判断する手がかりを提供し, 協調作業の状態推定に有効であると考えられる.

4.2 関連研究

4.2.1 プログラミング導入教育におけるペアプログラミング

既存研究では, プログラム作成課題を個人プログラミングとペアプログラミングで行わせたときに, 後者の有効性について述べている. 例えば, コードの質が高かった[McDowell, 2002][Hanks, 2004], 単位修得率が高かった[McDowell, 2002][Nagappan, 2003], 中間および最終試験の成績が高かった[Nagappan, 2003], 課題の提出率が上がった[Rountree, 2005]という結果が報告されている. また, ペアプログラミング実施前と実施後を比較したときについて, Urness らはコード理解やコード作成スキルがあがったと報告している[Urness, 2009].

いずれの研究もペアプログラミングの有効性を述べているが, プログラム作成課題中の過程や, 課題解決がうまくいかなかったペアについては言及していない. 本研究では, プログラミング導入教育におけるペアプログラミング学習において, ペア同士のやり取りを観察し, 課題の解決に失敗したペアについても言及している.

4.2.2 ペアプログラミングにおけるペア同士のコミュニケーション分析

プログラミング導入教育を対象とした研究ではないが, 作業中のペア同士のコミュニケ

ーションを分析している研究がある。Chen らは、作業中のペア同士の発話内容と発話時の状況を書き下し、発話内容から Driver と Navigator には精神的な距離があること、作業中に何らかのコミュニケーション支援が必要であることを示している[Chen, 2007]。Chong らも、作業中のペア同士の発話内容と発話時の状況を書き下し、ペアが所有する専門的な知識が作業中のインタラクションに影響する可能性があること、キーボードの操作権 (Keyboard control) がペアの意思決定に影響する可能性があることを示している[Chong, 2007]。Bryant らは、作業中の発話プロトコル分析を行って Driver および Navigator の発話内容の分布を調査し、両者の分布に差はなく、同じレベルで相互作用していることを示している[Bryant, 2008]。

文献[Chen, 2007]については、プログラム作成課題中にうまくいかなかった事例を示しているとも考えることができるが、いずれの研究でも課題解決に成功した事例と失敗した事例を比較することはしていない。本研究では、ペアプログラミングにおけるペア同士のやり取りを観察し、課題解決までのコミュニケーション活動について定量的に分析し、課題解決に成功した事例と失敗した事例を比較して、課題解決がうまくいかない事例の特徴について言及している。

4.2.3 ペアプログラミングにおける会話の役割

Wray[Wray, 2010]はペアプログラミングにおける会話の役割や会話をもたらす効果を実体験から述べている。この文献によると、ペアプログラミングにおける会話の役割は、ペア間で専門的知識を共有することや、手詰まりになったときに会話をすることによって、それを解決する手がかりを得ることと述べている。また、プログラムについて会話をするプログラマほどより生産的になり、また随時質問を相互に投げかけることは、何よりも生産的になると予測している。

これはつまづきが発生したときに会話をすることによって、つまづきを解決できる可能性があることを示している。本研究では、ペアプログラミングにおける成功事例と失敗事例を比較するが、会話がそれらを比較する指標になると考え、作業中の会話に着目して比較する。

4.3 ペアプログラミング演習

4.3.1 演習設定

本研究では、大学の情報系学部 1 年生を対象に開講された「プログラミング I」の講義内においてペアプログラミングを実施した。この講義では C 言語を題材とする教育が行われ

た。学習目標は次の 3 つである。

- ソフトウェアの記述・構成とプログラミングの仕組みが分かる
- C プログラミングのコンパイルと実行ができる
- C 言語の基礎（データ型，配列，制御構造，関数，文字列，ファイル）が分かる

講義は 2010 年 9 月から 1 回 75 分で計 10 回行われ，ペアプログラミングは講義時間の一部を使って実施した。演習は 6 回行われた。

演習を行うための準備として，演習参加者の C 言語プログラミング経験を問うアンケートを実施した。プログラミング I の学習内容程度を扱うことができると回答した者はプログラミング経験者とした。また，学習内容のすべては扱えないと回答した者はプログラミング初学者とした。初めてペアプログラミングを行う学生がいることを考慮して，本演習の前にペアプログラミングの練習を行った。

本演習では，それまでに学習した内容を題材としたプログラムを作成する課題を与え 30 分程度で完了できるようにした。課題例として図 4-1 に演習第 1 回の課題を示す。各回の演習について，実験者から演習参加者に対して次の指示を出した。

- このペアプログラミングでキーボードやマウスを操作できるのはコンピュータの手前に座っている Driver のみです。Navigator はキーボードやマウスに触れないようにして Driver をサポートしてください。触れなければどのようにサポートしても構いません。ディスプレイへ指差ししても構いません。
- プログラムの実行結果を確認し，課題の答えがでたら課題終了です。課題はできるだけ早く終了させてください。
- Driver, Navigator とも教科書[高橋, 2007]は閲覧しても構いません。Web ページの閲覧はしないでください。
- 教員および TA (Teaching Assistant) は，演習中は課題に対する質問は受け付けません。機器の不具合や操作方法に関する質問があるときのみ呼んでください。
- コメント文を使い，人間が見て見やすいものを作るように心がけてください。
- 制限時間は開始の合図から 30 分間です。時間になったら課題が途中の状態でも，できたところまでのコードを提出してください。

演習参加者は 62 名（学部 1 年生 52 名，2 年生以上 10 名）であった。コミュニケーションのしやすさ，ペアプログラミングのやり易さ，プログラミング経験差が発生することを考慮して，ペアは実験者が決定し，全ての回を通して同じペアを組ませることはしなかった。なお，参加者には誰がプログラミング経験者，プログラミング初学者なのかということとは伝えていない。ペアプログラミングの定義[Beck, 1999]に従うならば，1 回の演習において，Driver と Navigator の役割を交替させることもできるが，制限時間が 30 分と短いことから，1 回の演習では，Driver と Navigator のどちらか一方の役割を担当させることとした。演習は全 6 回であるが，全ての参加者について，Driver と Navigator の役割を 3

課題番号 1 整数型変数 n と r に $n \geq r > 0$ となる整数値を入力する。このとき、異なる n 個のものから r 個をとって 1 列に並べる順列の数を表わす nPr と、異なる n 個のものから r 個をとる組み合わせの数を表わす nCr を計算するプログラムを作成せよ。例えば n に 8, r に 3 を入力したとき、
 $8P3$ は 336 で $8C3$ は 56 です
 などと出力できればよい。 nPr と nCr はそれぞれ、

$$nPr = \frac{n!}{(n-r)!} \quad nCr = \frac{n!}{r!(n-r)!} = \frac{nPr}{r!} \quad \text{ただし、} n! = \begin{cases} n \times (n-1)! & (n \geq 2) \\ 1 & (n = 1) \end{cases}$$
 とする。

図 4-1 ペアプログラミング演習で用いた課題の例



図 4-2 データ取得用カメラの設置位置



図 4-3 ペアプログラミング演習で撮影した映像データの一場面

回ずつ経験できるようにペアを組んだ。ペアの組み合わせに優劣がでる可能性を考慮して、プログラムの良し悪しや課題完了までの時間等について優劣をつけることはせず、課題解決の途中であってもプログラムを提出した場合に一律の出席点を与えることを参加者に伝えた。演習環境について、演習参加者はプログラムを作成するエディタとして emacs を利

用している。

図 4-2 は演習風景，図 4-3 は設置したカメラで撮影したデータの一場面である。撮影は 1 組のペアに対して 3 台のカメラを使用し，演習の妨げにならないようにカメラを設置した。撮影は課題の開始から終了まで行い，ペア同士のやり取りを記録した。

4.3.2 データ収集

4.3.1 項における設定のもと，1 回の演習において 4 ペアの撮影を行い，6 回の演習でのべ 24 ペア分の会話を収集した。1 組のペアに対して 3 台のカメラを用いて撮影したが，その 3 台のカメラすべてで映像が鮮明かつ演習開始から演習終了まで撮影できている 10 ペアを対象として，つまずきの解決に成功した事例および失敗した事例を抽出した。

本研究におけるつまずきとは，「コンパイル時にエラーが発生した（コンパイルエラー）あるいはコンパイル時にエラーが発生しなくとも実行時にペアが意図した出力がなされなかった（実行時エラー）こと」と定義する。つまずきの解決に成功した事例（成功事例）とは，「コンパイルエラーに関するつまずきが発生した後にその後のコンパイルにて当該エラーが解決された事例，実行時エラーに関するつまずきが発生した後にその後の実行にて当該エラーが解決された事例」と定義する。つまずきの解決に失敗した事例（失敗事例）とは「つまずき発生した後に，演習の制限時間が来たために解決まで至らなかった事例」と定義する。後述の事例では，成功事例，失敗事例に関わらず事例中に複数回コンパイルエラーや実行時エラーが発生している場合もあり，これらのエラーが発生した時点では解決に失敗しているとも考えることもできる。しかし，このコンパイルやプログラムを実行した理由は，プログラムを少し修正してエラーが解決したかを確認したり，プログラムの途中に `printf` 文を挿入して途中の変数値を確認したりするためであり，このようなものは解決に失敗したのではなく，解決の途中段階として解釈した。

この定義にしたがって成功事例および失敗事例を抽出したところ，10 ペア全体で成功事例は 19 件（コンパイルエラー 12 件，実行時エラー 7 件），失敗事例は 3 件（コンパイルエラー 1 件，実行時エラー 2 件）あった。この 22 件について，分析結果の信頼性を損なわないようにするため，「つまずき発生後，ペア間で会話がないまま `Driver` が解決した（成功事例 7 件）」，「制限時間を超えてからエラーが発生した（成功事例 4 件）」，「`Navigator` がキーボード等を操作した（成功事例 1 件）」，「設置したカメラの話等の実験環境に関する話が会話の話題になる場面があった（成功事例 1 件）」事例を分析対象から除外した。以上の結果，成功事例 19 件のうち 6 件，失敗事例 3 件の計 9 件を対象とし，これらについて分析を行った。

表 4-1 に分析対象とする事例を示す。表 4-1 では成功事例 A～F の 6 例，失敗事例 A～C の 3 例を挙げている。つまずきは作業中にいくつか発生するため，1 組のペアに対して複数

表 4-1 ペアプログラミング演習中に発生したつまずきの事例

事例	ペア	Driver	Navi.	つまずき
成功 A	ペア A	初学者	経験者	コンパイルエラー 行末にセミコロンをつけ忘れていた.
成功 B	ペア B	初学者	経験者	コンパイルエラー 「enum」という文字列が予約語であった.
成功 C	ペア B	初学者	経験者	コンパイルエラー ソースファイルを上書き保存していなかった.
成功 D	ペア B	初学者	経験者	実行時エラー (セグメンテーションエラー) scanf 文で「scanf(“%d”,n);」と書いていた. &を忘れていたことが原因
成功 E	ペア C	初学者	初学者	コンパイルエラー ブロックの開始と終了が対応できていなかった. (このペアはテキストモードで編集しており, ブロックチェック機能が働いていなかった) プログラム内にスペリングミスがあった.
成功 F	ペア C	初学者	初学者	実行時エラー コンパイルはできるが正しい答えが得られなかった. 関数の返り値が正しく返されていなかった.
失敗 A	ペア D	経験者	経験者	実行時エラー コンパイルはできるが正しい答えが得られなかった. (繰り返し文内において, 零で除算する場合があった)
失敗 B	ペア A	初学者	経験者	実行時エラー コンパイルはできるが正しい答えが得られなかった. (グローバル変数宣言により変数値が不正だった)
失敗 C	ペア E	初学者	初学者	コンパイルエラー メインファイルとヘッダファイルがリンクしなかった.

※初学者：プログラミング初学者，経験者：プログラミング経験者 ※ペアはすべて1年生同士

表 4-2 ペアプログラミング演習で収集したコミュニケーションデータ

事例	データ長 (秒)	Driver の発話 数	Navigator の発話数	発話 時間 (%)	発話頻 度 (発 話回数/ 分)	発話 長(秒 /発話 回数)	説明 の繰 り返 し(%)	連続 発話 率(%)
成功 A	100	10	11	16.2	6.3	1.55	0	47.6
成功 B	151	24	27	31.2	10.1	1.85	0	31.4
成功 C	50	5	6	25.9	6.6	2.35	0	18.2
成功 D	125	14	15	21.3	7.0	1.84	0	17.2
成功 E	121	12	8	14.4	5.0	1.74	5.0	40.0
成功 F	417	35	23	12.6	4.1	1.81	6.9	32.8
失敗 A	624	82	53	26.5	6.5	2.45	5.2	28.9
失敗 B	588	19	61	12.9	4.1	1.89	8.8	62.5
失敗 C	460	63	53	33.8	7.6	2.68	7.8	29.3

の事例がある場合がある。表 4-1 に挙げられている事例は、最終的に当該ペアが認識したエラーの原因であり、つまずきが発生してからエラーの原因を認識するまでに多少の時間を費やしている。失敗事例 A と B について、つまずきの事例の括弧内は実験者が撮影したビデオ映像を観察することで分かったエラー原因であり、Driver および Navigator はそのエラーに気付いていなかった。

なお、本演習では失敗事例が 3 例しかなかった。これは本演習で演習参加者に与えたプログラム作成課題が、それまでに学習した内容を題材としたもので、30 分程度で完了できるものであることによるためだと考えられる。撮影したほとんどのペアは制限時間内に課題を完了させていた。

4.4 発話分析

4.4.1 データ処理

撮影した映像を用い、ビデオ分析ツール iCorpusStudio[iCorpusStudio]を使用して、ペアごとに Driver と Navigator の発話内容を記録した。iCorpusStudio は、同時に複数映像を見ながらタイムラインに対して複数種類のラベリングをし、ラベルのカテゴリごとに CSV 形式のファイルに出力できる。表 4-2 に取得したコミュニケーションデータを示す。成功事例のデータ長は表 4-1 で示した各つまずきが発生した時点からそれが解決されるま

での時間である。失敗事例のデータ長は、つまずきが発生した時点から参加者が解決を止めた時点（制限時間が来たために途中のコードを提出し始めようとした時点）までの時間の長さである。発話数は事例中に発声した回数である。本研究での発話とは、時間の長短、意味の有無によらず、1回の発声を指す。つまり、「あー」、「うん」等の短い相槌もすべて発話として数えている。説明の繰り返し、連続発話率、発話頻度、発話長、発話時間については次項で説明する。

4.4.2 分析方法

つまずきの解決の成功事例と失敗事例における発話の様子を調べるために、次の点に着目して分析した。

(1) 発話量

本研究では発話量について、発話時間、発話頻度、発話長を分析した。発話時間とはデータ長に対してどの程度発話が生じていたかを割合で示したものである。発話頻度とは1分あたりの発話数である。発話長とは1回の発話の開始から終了までの平均時間である。表4-2で示されている発話時間、発話頻度、発話長はDriverおよびNavigatorそれぞれで算出して平均化し、1人あたりの値を示している。

(2) 説明の繰り返し

一般に、聞き手が誤解している時に話し手はより強い声の調子や強調したジェスチャで話を繰り返す、あるいは、話し手が曖昧なことを言ったときに聞き手が聞き返しや困った表情をすることによって不理解を合図し明確化を求めるような状況がある。このような状況下では、話し手の一回の発話で聞き手がその内容を理解できずに話し手が説明を繰り返すことになる。一回の発話で聞き手が理解できない場合はスムーズなやり取りができていないと考えられる。説明の繰り返しとは、このように同じ内容を繰り返して発話することと定義する。

分析では、次の3つの状況が起きた回数を事例ごとに数えることとした。

- 話し手が発話した後に、聞き手が誤解していると話し手が感じ取り、話し手が説明を繰り返した
- 話し手が発話した後に、聞き手が何も反応しない（発話しない、何も操作しない）ときに、話し手が説明を繰り返した
- 話し手が発話した後に、聞き手が聞き返しをして、話し手が説明を繰り返した

説明の繰り返しの例として、表4-3に失敗事例Bで行われた会話の一部を示す。ここではwhile文を別の行に移動させようとしている。この会話の発話番号28においてNavigatorが「この関数は外に出してもいいから」と指示を出し、Driverが操作をするが、それを見たNavigatorはDriverが誤解していると気づき「あー、もう以下全部、中括弧閉じるまで」

と Navigator がやりたいことについて繰り返し説明している。これは話し手が発話した後に、聞き手が誤解していると話し手が感じ取り、話し手が説明を繰り返したという状況にあたる。このような状況が起きた回数を数えている。回数を数えたあと、発話数に対する説明の繰り返しの比率を算出した。

(3) ペアの両者が交互に発話できているか

話し手と聞き手が交互に発話を行うことで、会話はスムーズに行われていると考えることができる。例えば、話し手の説明に対して聞き手が相槌等の応答をすることにより、話し手は伝えたいことが聞き手に伝わったことを確認でき、聞き手は自身が話し手の発話内容を理解したと話し手に伝えることができる。交互に発話できていないと、一方的に話し手から聞き手に発話することになり、話し手は聞き手が理解できているか分からず、聞き手も話し手の発話内容を理解できていない可能性が高く、スムーズなやり取りができていないと考えられる。

分析では、各事例に対して Driver あるいは Navigator が連続的に発話した回数を数え、その割合を算出することとした。これを本研究では連続発話率と呼ぶ。連続発話率は、Driver と Navigator の会話全体に対して、どのくらい交互に発話できていないかを割合で示したものである。このことを、図を使って説明する。



図 4-4-1 Driver と Navigator が完全に交互に発話している事例

図 4-4-1 は、ある事例において、Driver と Navigator が事例中に何回発話してどの順番で発話されたかが示されている。図 4-4-1 は Driver が最初に発話を行い、2 番目に Navigator が発話し、3 番目に Driver が発話し、…、最後に Navigator が発話したという事例である。

ある発話の後、次の発話の話者が変わったとき「発話交替した」とする。図 4-4-1 では、1 番目の発話者が Driver であり、2 番目の発話者が Navigator なので、この 2 つの発話については発話交替したことになる。同様に、2 番目・3 番目の発話についても発話交替したことになり、3 番目・4 番目の発話についても発話交替したことになり、…、9 番目・10 番目の発話についても発話交替したことになる。

図 4-4-1 では 10 回発話されているが、1 番目・2 番目の発話について発話交替が 1 回、2 番目・3 番目の発話について発話交替が 1 回、…、と数えていくと、10 回の発話に対して合計で発話交替が 9 回あったことになる。このとき発話交替していないところはない。



図 4-4-2 Driver と Navigator が完全に交互に発話していない事例

図 4-4-2 は、ある事例において、Navigator しか発話しなかった場合を示す。図 4-4-1 の場合と同様にして発話交替があったかどうかを数えていくと、10 回の発話に対して合計で発話交替が 0 回あったことになる。このとき 9 回発話交替していないところがある。



図 4-4-3 Driver と Navigator が発話交替しているところと発話交替していないところがある事例

図 4-4-3 は、ある事例において、Driver と Navigator が発話交替しているところと発話交替していないところがある場合を示している。この事例においては、3 番目・4 番目の発話について発話交替を 1 回しており、4 番目・5 番目の発話について、7 番目・8 番目の発話について、8 番目・9 番目の発話について、発話交替を 1 回している。この事例では、10 回の発話に対して合計で発話交替が 4 回あったことになる。このとき 5 回発話交替していないところがある。

図 4-4-1 から図 4-4-3 を使って説明したように、Driver と Navigator の発話全体に対して、発話交替していない回数を数え、この数値を使って、どのくらい発話交替していないかを示したものが連続発話率になる。連続発話率を次のように定義する。

Driver の発話数を S_D 、Navigator の発話数を S_N 、発話交替していない回数を C とするとき、

$$\text{連続発話率} = \frac{C}{S_D + S_N - 1} \times 100(\%) \quad (\text{ただし、} S_D \geq 1, S_N \geq 1)$$

連続発話率の分母は、事例中の発話全体について、発話交替できる回数の最大値である。例えば、図 4-4-1 の例では Driver の発話数は 5、Navigator の発話数は 5 で、合計 10 回の発話をしているが、発話交替できる回数は最大で 9 回 (=10-1 回) である。これを踏まえると図 4-4-1～図 4-4-3 の例において発話交替できる回数はいずれも最大で 9 回である。

この発話交替できる最大の回数を分母として、発話交替していない回数を分子にすることにより、連続発話率は 0% から 100% の値をとることができる。図 4-4-1 の例では発話交替していない回数は 0 回なので連続発話率は 0%、図 4-4-2 の例では発話交替していない回数

は 9 回なので連続発話率は 100%，図 4-4-3 の例では発話交替していない回数は 5 回なので連続発話率は 55.5%となる（有効数字 3 桁，4 桁目以降を切り捨て）。

表 4-3 失敗事例 B で行われた会話の一部

発話 番号	話 者	発話 開始	発話 終了	発話内容
14	N	1345.0	1346.2	while 文を
15	D	1346.3	1346.6	うん
16	N	1348.4	1350.0	main の外に出そう
17	D	1351.0	1352.1	main の
18	N	1356.1	1357.1	上のほうでいいや
19	D	1357.7	1358.3	うん
20	N	1360.0	1361.7	うん，そこまでの間
21	D	1362.3	1363.5	うん
22	N	1364.2	1366.7	while 文で，切り取りで
23	N	1369.7	1370.4	でー
24	N	1371.4	1372.6	えーっと
25	N	1373.4	1374.6	while 文 (Driver が操作)
26	N	1378.0	1380.8	あつ，while 文じゃない，あ，ごめん，元に戻して
27	N	1384.9	1387.0	ごめん，おかしくなった
28	N	1387.5	1391.8	この関数は外に出してもいいから (Driver が操作)
29	N	1393.0	1396.4	あー，もう以下全部，中括弧閉じるまで (Driver が操作)
30	N	1407.9	1413.5	でも出したから，これで宣言，漏れたらあぶない
31	N	1416.0	1420.1	jyun（変数名）は大丈夫なんだよ，ans も大丈夫，n は
32	N	1422.0	1423.2	ああ，n は
33	N	1431.7	1433.6	関数の外で宣言していいんだっけ
34	N	1437.9	1440.9	グローバルなんちゃらってある
35	N	1442.3	1443.1	索引で
36	N	1446.0	1450.5	グローバル，グローバル変数
37	N	1456.0	1458.0	あつ，ブロックの外いけるや

※話者：D は Driver，N は Navigator，発話開始と発話終了は発話開始・終了時間（単位：秒）

4.5 分析結果

本節では 4.4 節で示したデータおよび分析方法に従って、成功事例と失敗事例を比較した結果を示す。統計処理は SPSS Statistic 17.0 を用いて行い、比較は Mann-Whitney の U 検定を利用した。この検定は 3 章で述べた t 検定と同じように、独立な 2 組の標本に有意差があるかどうかを検定する手法である。Mann-Whitney の U 検定では、その母集団の分布などの前提を必要としないため、汎用性のある手法である。後述の Mann-Whitney の U 検定では両側検定を行った。

4.5.1 発話量

(1) 発話時間

図 4-5 は両事例全体でどの程度発話が生じていたかを割合で示したものである。以下エラーバーは標準偏差を表す。成功事例の平均発話率は 20.4%，失敗事例では 24.4%であった。Mann-Whitney の U 検定で両事例の発話時間を比較したところ、有意水準 5%のもとで有意な差があるとは認められなかった ($U=6$, $p=0.439$)。よって、成功、失敗によって発話時間に差があるとはいえない。

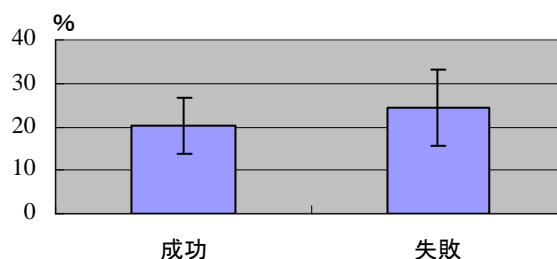


図 4-5 1人あたりの発話時間

(2) 発話頻度

図 4-6 は両事例における発話頻度を示している。成功事例の平均発話頻度は 6.51 回、失敗事例では 6.06 回であった。Mann-Whitney の U 検定で両事例の発話頻度を比較したところ、有意水準 5%のもとで有意な差があるとは認められなかった ($U=8.5$, $p=0.897$)。よって、成功、失敗によって発話頻度に差があるとはいえない。

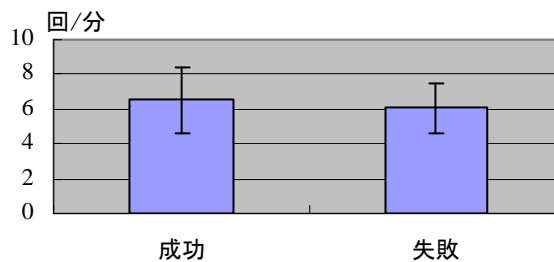


図 4-6 1人あたりの発話頻度

(3) 発話長

図 4-7 は両事例における発話長を示している。成功事例の平均発話長は 1.85 秒，失敗事例では 2.34 秒であった。Mann-Whitney の U 検定で両事例の発話長を比較したところ，有意水準 5%のもとで有意な差が認められた ($U=1$, $p=0.039$)。これにより失敗事例の発話長が成功事例よりも長いことが分かった。

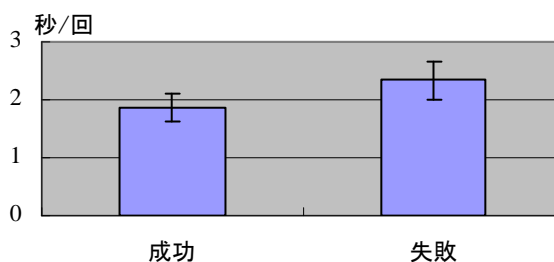


図 4-7 1人あたりの発話長

4.5.2 説明の繰り返し

図 4-8 は両事例における説明の繰り返し比率を示している。成功事例の平均比率は 1.98%，失敗事例は 7.26%であった。Mann-Whitney の U 検定で両事例の説明の繰り返し比率を比較したところ，有意水準 5%のもとで有意な差が認められた ($U=1$, $p=0.031$)。失敗事例の説明の繰り返し比率は成功事例よりも大きいことが分かった。

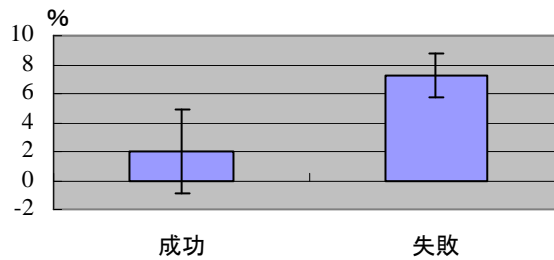


図 4-8 説明を繰り返した回数（発話数に対する比率）

4.5.3 ペアの両者が交互に発話できているか

図 4-9 は両事例における連続発話率を示している。成功事例の平均連続発話率は 31.2%、失敗事例では 40.2%であった。Mann-Whitney の U 検定で両事例の連続発話率を比較したところ、有意水準 5%のもとで有意な差があるとは認められなかった ($U=8$, $p=0.796$)。よって、成功、失敗によって連続発話率に差があるとはいえない。

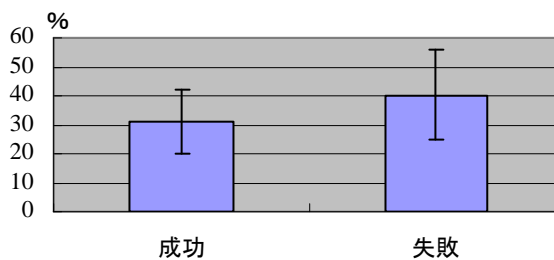


図 4-9 連続発話率

4.6 考察

本研究では、ペアプログラミング学習におけるペア同士のコミュニケーションデータから、Driver と Navigator の会話を定量的に分析することで、作業中に発生するつまずきの解決に成功した事例と失敗した事例の違いを示している。本章では、4.5 節で得られた結果について検討を行い、協調作業支援に与える示唆を論じる。

4.6.1 作業中のペア同士のやり取り

説明を繰り返した回数、発話長について有意差が認められ、失敗事例のほうがそれらが

大きいことが認められた。連続発話率について有意差は認められなかったものの、失敗事例のほうが成功事例に比べて約 9%連続発話率が高いことが分かった。

このことから、つまりきを解決できないことが発話長等に現れていることが分かった。表 4-3 の事例ではそれが顕著に現れている。発話番号 14 から 22 までは Driver と Navigator がほぼ交互に発話をしており、両者のやり取りはできていると考えられる。しかし、発話番号 22 から発話番号 37 まで Navigator が連続して発話しており、かつ 1 回の発話長が 2 秒以上ある発話数は 10 と多いと考えられる。4.4.2 項で説明したように説明の繰り返しも行われている。この間 Driver は Navigator の指示どおりにコンピュータを操作していた。しかし、発話番号 29 に対して Driver がコードを修正した後、しばらく Driver は発話を行わず、じっと画面を見ているだけであった。発話番号 29 までは、Driver は Navigator が伝えたいことを理解していたと考えられるが、その後 Navigator が連続して発話長の長い発話をしてしまったために、Driver が混乱をおこしてしまい、Navigator に対して何も言えなくなったと考えられる。そこにグローバル変数という新たな話題が提示されるが、Driver はそれまでの Navigator からの発話内容を消化できずに新たな話題に移行してしまい、さらに混乱が起きてしまったと考えられる。この後、演習の制限時間までグローバル変数に対応しているが、Driver は Navigator が言ったとおりに操作することに終始しており、つまりきの解決には至らなかった。

このように発話長の長さ、説明の繰り返し、連続発話をきっかけとして、Driver と Navigator のやり取りがうまくいかず、聞き手が話し手の内容をきちんと理解できなかった、あるいは混乱してしまった可能性がある。その結果、失敗事例ではつまりきの解決まで至らなかったのかもしれない。しかし、表 4-3 の例では、これらの 3 要素が連鎖的に発現してしまったために、両者のやり取りがうまくいかなかった可能性がある。それぞれが単独で発現した場合にどのようなやり取りが発生するかについては、今後データ数を増やす等をして検討していく必要はある。

一般に、個人の能力や授業内容の理解度には差があると考えられ、例えばプログラミング経験を指標として課題の成功と失敗を判断しようとしても、それが難しいであろうことは、表 4-1 の事例においてプログラミング経験によらず成功や失敗がみられることからわかる。本研究では、成功事例と失敗事例を比較する指標として、ペアの会話に着目したところ、成功と失敗の差が会話に表れていることが分かった。

4.6.2 本研究の限界

本演習では、講義の一部の時間を使ってペアプログラミング演習を行ったために、演習時間が毎回 30 分となった。Beck らが提唱したペアプログラミングの定義に従うならば Driver と Navigator の役割を交換することもできたが、時間の関係で各回の演習において、どちらか一方の役割を担当するというようにした。当然ながら、演習中に役割を交換する

ことで、会話内容に変化が見られ、つまずきの解決がさらにスムーズに進む可能性も考えられる。しかし、この制約は演習参加者に等しく与えられており、本研究ではその制約の中で、つまずきの解決に成功した事例と失敗した事例を比較している。

本演習で出したプログラム作成課題は、それまで学習した内容で、30分程度で完了できるものであることから、難易度は低めに設定されていた。これは課題を完了できないと、学習に対するモチベーションが下がってしまうことを懸念したためであるが、そのために、ほとんどのペアが課題を完了させており、失敗事例が少なかった。課題の難易度を上げることで、本演習よりもつまずきが多く発生すると考えられ、その結果多くの失敗事例データを得ることができると考えられる。これについては、今後さらに検討していく必要がある。

4.6.3 協調作業の状態推定

本研究の結果から例えば、ペアプログラミング学習において、ある話し手の発話長が平均よりも長く、話し手が一方的に発話していると、ペアプログラミングがうまくいっていないのではと推定することができるであろう。さらに、ペアプログラミングは協調作業であることから、より一般に協調作業の状態推定にも有効である可能性はある。

本研究における演習では、同学年同士のペアかつ講義開始が9月であったことから、ペア同士のコミュニケーションは比較的容易に行うことができたと考えられる。つまり、会話のやり取りがうまくいかず、聞き手が話し手の発話内容が理解できなくても容易に聞き返すことができた、あるいは、話し手が容易に聞き手に何らかのサポートをすることができたであろう。このために失敗事例が少なかったのかもしれない。しかし、仮に同学年同士ではない、あるいはペアが初対面であった場合には容易にコミュニケーションをとることはできない。本研究ではペアプログラミングにおいて、会話に関するいくつかの指標が作業の状態と関係があることが分かったが、これらの指標がより一般的な協調作業と関係があるということが分かれば、会話を作業状態の指標として使うことができるであろう。

このような可能性は、今後データ数を増やす等をして、さらに検証をしていく必要があるが、本研究の結果は協調作業をどのように支援していけばよいかを検討する一つの手掛かりとなるであろう。

4.7 まとめ

本研究では、プログラミング導入教育で実施したペアプログラミング学習において、作業中に発生したつまずきの解決に成功した事例と失敗した事例をペア同士の会話の観点から定量的に分析比較した。

その結果, つまづきの解決に失敗した事例について, 成功した事例と比較したときに, (1)1回の発話長が長い, (2)説明を繰り返す回数が多い, (3)一方的に発話する回数が多いことが分かった.

今回得られた知見に基づいて, より一般的な協調作業の状態を推定することが可能になれば, これまで以上にペアプログラミング学習や協調作業の効果が得られるようになるであろう.

第5章 総合検討

2.2.3 項で述べた問題点を解決するために、本研究では分散環境における協調プログラミング学習と対面環境における協調プログラミング学習を実施した。本章では、既存研究で得られた知見と本研究で得られた知見を踏まえ、分散環境における協調プログラミング学習支援と対面環境における協調プログラミング学習支援という観点で検討を行う。その後、本研究では明らかにできなかったことを今後の課題として述べ、最後に、今後の協調プログラミング学習支援について、将来の展望を述べる。

5.1 本研究で得られた知見のまとめ

第3章では、分散環境における協調プログラミング学習として、作問学習支援システムを利用して実施した、分散非同期環境での協調プログラミング学習を調査・分析した。その結果、次の3点が分かった。

- システムを積極的に利用すること、学習者が作成した問題に多く解答すること、学習者が作成する問題の正確性が高いことが、成績向上に関係している
- 問題に解答することが成績向上に大きく関係している
- 正確性の低い問題を作成することが成績低下に大きく関係している

既存研究において、分散環境における学習を支援している研究は、Wiki やデータベースに Web コンテンツを蓄積し、それを閲覧できるようにすることによって、距離的に離れている学習者同士で学習内容を共有することを可能とするものであった。第3章で実施した内容もデータベースに「学習内容に関する問題」という Web コンテンツを蓄積し、それを閲覧できるようにしたものである。本研究では、それに加えて作問学習という学習方法を取り入れた協調プログラミング学習支援を実施し、その結果を示した。

第4章では、対面環境における協調プログラミング学習として、ペアプログラミング学習を分析した。その結果、プログラミング中に発生するつまずきの解決に失敗した事例について、成功した事例と比較したときに、次の3点が分かった。

- 1回の発話長が長い
- 説明を繰り返す回数が多い
- 一方的に発話する回数が多い

既存研究において、プログラミング中の学習不振者に関する考察が十分行われていなく、本研究ではその一考察として、協調プログラミング学習中における会話と、発生するつまずきの解決の成功・失敗との関係を考察した。

5.2 検討

本節では、既存研究と本研究で得られた結果を統合して、分散環境および対面環境における協調プログラミング学習支援について検討する。検討は、支援方法、学習者の振る舞い、支援効果の 3 つの観点で行う。支援方法については、既存研究や本研究における協調プログラミング学習の支援において、どのような支援が行われたのか、また、その支援と協調学習との関わりやその支援方法の意義について考察する。学習者の振る舞いについては、既存研究や本研究における協調プログラミング学習の支援において、学習者がどのような振る舞いをしたのかについて考察する。支援効果については、既存研究や本研究における協調プログラミング学習の支援において、どのような効果が得られたのか、また、それは今後の支援においてどのように役立つのかについて考察する。

5.2.1 分散環境における協調プログラミング学習支援

(1) 支援方法

Wiki やデータベースの利用

既存研究や本研究における支援では、Wiki やデータベースにプログラミングに関する Web コンテンツを蓄積し、それを学習者が閲覧できるようにすることによって、距離的に離れている学習者同士で学習内容を共有することを可能としている。これにより、ある学習者がプログラムを作成する際に、他学習者が作成したプログラムを閲覧しながら、自身のプログラムを作成することができる。これは、観察学習 (Observation Learning) を実現しており、学習者は自分で問題を解決する体験を通じてのみならず、他者の問題解決行動や振る舞いの観察を通じて間接的にも知識やスキルを獲得することが可能となる。

自身が作成しているプログラムと同じようなプログラムを閲覧するのであれば、Wiki 等を利用しなくとも、膨大な Web ページの中から自分と同じようなプログラムを見つけて、それを見ながらプログラムを作成していけばよいこともある。しかし、膨大な Web ページの中から自分に合うプログラムを探すことは、場合によっては困難なこともある。Wiki 等を利用することで、自分が行っている問題解決行動と同じことを行っている学習者のプログラムを閲覧することができるため、膨大な Web ページの中から探すよりも、時間的に効率よくプログラムを閲覧することができるという利点もある。

コミュニケーションの場の提供

Wiki ページや支援システム内に掲示板や Q&A ページを提供している場合がある。これにより参加者が意見交換、競合、交渉、合意形成等を行うことを可能としている。参加者

同士でコミュニケーションを行うだけであれば、今日では Skype や Windows Messenger といったコミュニケーションツールがあり、それを利用すればよい場合もある。

本研究でも支援システム内に掲示板を設置した。表 3-2 にあるように 2008 年度では 23 件、2009 年度では 9 件の投稿と、その他の利用状況と比較すると、掲示板は利用されていないように考えられる。本研究では掲示板を利用することでシステムが学習者に何か支援をするということはしてはいてなく（投稿した内容に返信があった場合にそれを支援システムのトップページで知らせる機能はある）、また、掲示板が利用されなかった原因については、十分な考察はできていない。

今後の支援環境において、例えば、掲示板や Q&A ページに投稿された内容に応じて、システム等が、投稿した学習者に何らかの支援をするということを考えるのであれば、それらを提供することには意義があると考えられる。

学習に対するモチベーション向上の支援

分散環境における e ラーニングでは、ドロップアウト者がでる[山田, 2010]とされているように、分散環境における学習では対面環境における学習と比較すると緊張感等が落ち、学習に対するモチベーションが下がることもある。このために、支援環境においてモチベーションを支援している場合がある。

本研究でも、3.3.2 項で述べたように作問ポイントと貢献ポイントに応じたランキングを表示することで、モチベーションの向上を図った[平井, 2008]。このランキングによるものかどうかは分からないが、3.5.3 項のアンケート回答に「私だけでなく、他の人もたくさん勉強しているのが見え、自分も負けるものかと思い、やる気になりました」等とあるように学習に対するモチベーションが向上していることが伺えた。

モチベーション向上の支援と協調学習との直接的な関わりはないと考えられるが、今後の支援では、学習に対するモチベーションの向上についても考慮する必要があると考えられる。

作問学習

分散環境における作問学習を利用した協調プログラミング学習は、著者が知る限り、本研究のみで行った実践である。そのため、本研究の結果のみから考察することとなるが、今後の支援に向けて、重要な知見となるであろう。

2.1.3 項で述べたように、プログラミングに関する問題に取り組むことで、プログラミング学習支援をする手法も提案されている。そこでは、システムが自動生成した穴埋め型問題に解答すること[梶田, 2009]や、ソースコードから作成した穴埋め式のドリルをシステムが自動作成する[内田, 2007]というものであり、与えられた問題を学習者が解くという学習方法であった。本研究では、その問題を学習者自身が作成し、かつ作成された問題を評価するというを加えた。3.1 節や 3.2 節で述べたように、問題を作成することや問題を評

価することに良い効果があることは、多くの既存研究で言われており、ただ単に問題を解くよりは学習的に良い効果が得られると考えられる。

作問学習を利用した協調プログラミング学習は、いくつかの点で、協調学習の有効性を示唆する理論的な背景や現象に即した学習といえる。

- **Peer Tutoring:** 作成された問題に解答し、その問題を評価するという活動に **Peer Tutoring** が含まれる。問題評価活動には、問題文や解答・解説が正確に作られているかどうかという評価を、作問者とは別の学習者が客観的に行う。この評価では、問題文や解答・解説の誤りを指摘することもある。正確性を評価するあるいは誤りを指摘するためには、当該学習内容を理解している、あるいは当該学習内容を復習する必要がある。つまり、問題評価活動では、作成した問題を評価することで問題を作成した学習者を助け、かつ、誤りを指摘することで当該学習内容を学習することができる。
- **Observation Learning:** 作成された問題を閲覧するという活動に **Observation Learning** が含まれる。他学習者が作成した問題を閲覧することにより、どのようにプログラムを作成したのか、あるいは学習内容をどのように応用してプログラムを作成したのか等を観察することが可能となる。つまり、作成された問題を閲覧するという活動では、学習者が自分で問題を作成する体験を通じてのみならず、他学習者が作成した問題の観察を通じて、当該知識を獲得することができる。
- **Cognitive Apprenticeship:** **Cognitive Apprenticeship** は、他者が新しい知識やスキルを現実的問題に適用する過程を学習者が観察し、ガイドを受けながら実際に自分で行ってみることによって新しいスキルを獲得していく過程である。これを作問学習にあてはめることができる。例えば、他者が新しく作成した問題を観察し、そのような問題を自分で作成し、さらに、作成した問題に対する評価（問題を修正するためのガイド）を踏まえて、問題を修正していくことで、新たな知識を獲得することができる。
- **Legitimate Peripheral Participation:** **Legitimate Peripheral Participation**（正統的周辺参加）は、あるコミュニティにおける新参者としての学習者が、コミュニティのメンバとともに問題解決し、課題を達成していく過程を通じて関連する知識やスキルを獲得し、やがてコミュニティの十全的参加者へと成長していく過程である。これを作問学習にあてはめることができる。本研究では見られなかった現象ではあるが、例えば、支援システムを利用している学習者コミュニティに新たに加わる学習者が、新たな問題を作成し、その問題についてコミュニティに既存の学習者が評価を行う。その後、新たな学習者が問題を修正あるいは完成させることで当該学習内容に関する知識を得ることができ、やがてコミュニティの十全的参加者へと成長する過程もあると考える。

作問学習を利用したプログラミング学習は分散環境における支援に限った手法ではない。

今日も作問学習に関連する研究は多く行われており、今後普及する学習方法の 1 つであると言えよう。今後の支援においては、作問学習を取り入れた協調プログラミング学習支援を考案することも良いと考える。

(2) 学習者の振る舞い

分散環境における協調プログラミング学習において、学習中の学習者の振る舞いを考察しているのは、筆者の知る限り本研究のみである（黎らの研究[黎, 1998]では、分散環境におけるグループプログラミング教育支援システムを提案しているが、そのシステムを利用した実践分析は行っていない）。

本研究で開発した支援システムにおける学習者の振る舞いについては表 3-2 で支援システムの利用状況を示し、表 3-3 で支援システムに登録された問題とその問題に設置されている BBS で行われた議論の例を示した。表 3-3 の例を見ると、作問者に対する質問とその回答をベースとして、作問者・質問者とは別の学習者が、異なる観点から説明をした結果、作問者も含めて複数の学習者が問題の解説内容を理解したという流れになっている。これによりプログラミング学習ができたかどうかを断定することはできないが、学習内容を理解するプロセスの一例を示すことができたと考える。

分散環境での協調プログラミング学習における学習者の振る舞いについては、まだ検討する余地が残っており、今後も継続して調査・分析していく必要があろう。

(3) 支援効果

分散環境における協調プログラミング学習に限ったことではないが、学習プロセスに支援効果があったかどうかを定量的に評価することは難しい。そのためかどうかは断定できないが、支援環境利用後（プログラミング学習終了後）に実施したアンケートの回答内容やプログラミング学習に関する試験結果から支援効果について考察することが主流となっている。

支援効果を考察している中尾らの研究[中尾, 2006]では、授業後のアンケートの分析から、情報共有と協調学習ツールの有用性の認識およびプログラミング学習への意欲との関連性が示唆されたと報告している。また、谷口の研究[谷口, 2005]では、支援環境に利用後のアンケートの分析から、プログラミング学習において作品の作成過程における情報を学生が共有することにより、協調的かつ効果的な学習が行われたと報告している。これらの結果は **Observation Learning** による効果も示していると考えられる。

本研究では、支援効果を定量的に明らかにするために、支援システムの利用状況と試験の成績を利用して分析した。3.8.4 項で述べたように、支援システムにおける問題解答数の多少が成績の向上、低下に関係していると考えることができた。問題を解くことに学習効果があることは多くの既存研究で明らかにされており、また、プログラミングに関する問題を解くことによる効果も明らかにされている[梶田, 2009][内田, 2007]。本研究で得られた

結果は、問題を解くことに効果があるという事実を改めて示しているといえる。本研究では、この事実に加えて、作問学習特有の事象である、問題を作成することによる効果、作成する問題の正確性の高さによる効果を示した。

本論文では、分散環境における協調プログラミング学習支援として、既存研究で行われた Wiki やデータベースを利用した学習支援や、作問学習活動を利用した学習支援を分析し、その結果を示した。それぞれの学習支援において、一定の効果が得られており、協調プログラム学習支援として有効であると考えられる。分散環境における協調プログラミング学習支援に関する実践例はまだ少ないが、今後、分散環境における学習が普及するにつれて、実践例も増えていくだろう。今後も継続して実践例や実践の結果に注目し、それらを今後の協調プログラミング学習支援に生かしていくことが必要であろう。

5.2.2 対面環境における協調プログラミング学習支援

(1) 支援方法

グループによる学習

個人単位で行う学習をグループ単位で行うことにより、個人単位の学習では得られなかったことを得る機会がある。既存研究では、主に他者評価[新開, 2008]やピア・レビュー[生田目, 2004][北, 2007]のように、ある学習者が作成したプログラムやフローチャートを評価あるいは誤りを指摘する等の活動が行われている。これは Peer Tutoring あるいは Observation Learning を実現した例である。他学習者が作成したプログラムやフローチャートを観察し、必要に応じてそれらの誤り等を指摘することで、グループ内で相互に教えあい、自ら不足している知識を補強しながら課題に取り組むことができる。作成したプログラムやフローチャートを自身で客観的に評価することは難しく、また、他学習者からの助言等によって、初めて得られる知識もある。

ペアプログラミングを利用した学習

ペアプログラミングを利用した学習もグループ単位で行う学習であるが、ペアプログラミングでは Driver と Navigator の 2 人 1 組で行うため、参加者のコミュニケーションが密に行われる。

2.1.4 項で述べたように Preston は、ペアプログラミングと協調学習の関連を「Common task or learning activity suitable for group work」, 「Small group learning」, 「Cooperative behavior」, 「Interdependence(or positive interdependence)」, 「Individual accountability and responsibility」の 5 つの観点で示し、ペアプログラミングを利用した協調学習が有効であることを示している[Preston, 2006]。2.1.2 項で述べた協調学習の有効性を示唆する理論的背景や現象を踏まえるのであれば、いくつかの点で、それらに即した学習といえる。

- Peer Tutoring: Navigator が Driver のプログラムを観察して、プログラムの誤りを

指摘すること,あるいは,作成するプログラムの指針を与えること等はPeer Tutoringを実現した例であると考えられる. また, Driver が作成したプログラムについて, どういう意図でこのプログラム(プログラムの行, プログラムの文等)を書いたのかということを Navigator に説明することも Peer Tutoring を実現した例であると言える. これらの例が実践されることで, 学習者が相互に助け合い, 相手に教えることによって自ら学習することも可能となる. ペアプログラミングでは, Driver や Navigator の交換もできるために, お互いが Driver, Navigator になることができ, それぞれの役割における効果を得ることができる.

- **Observation Learning:** 主に, Driver が作成するプログラムの作成過程を Navigator が観察することで Observation Learning が実現されている. Driver が作成するプログラムの作成過程を Navigator が観察することにより, プログラムの書き方や学習内容をどのように応用することができるか等の知識やスキルを得ることが可能となる.
- **Cognitive Apprenticeship:** Driver が新しく得た知識やスキルを使ってどのようにプログラムを作成するのかというプログラム作成過程を Navigator が観察することに加え, Driver と Navigator を交換することによって, そのプログラム作成の続きを, ガイドを受けながら行うことができる. これは Cognitive Apprenticeship を実現した例であると言える.
- **Self-regulated Learning:** Self-regulated Learning では, 学習者が現在行っている課題の内容とは別に, 現在自分が採用している戦略が効果的か, またプラン通りに実行されているかなど, 認知的ゴール達成に関する評価を行う. ペアプログラミングを提唱した Beck の研究[Beck, 1993]によれば, Navigator の役割には「プログラムの作成方法をより戦略的に考える」ことも含まれると述べており, これは Self-regulated Learning にあたると考えることができる.
- **Sociocultural Theory:** 学習者には, 個人で実行可能な領域と実行不可能な領域があり, 実行不可能な領域の中には, 他者の援助を得れば実行可能な領域が存在する. つまり, 発達段階, 知識獲得段階の近い学習者同士をペアにして相互作用させることによって, 適した領域の機能が他者との間で実現され, それが学習者に内化される. 本研究で実施したペアプログラミングは, プログラミング導入教育場面で行われたものであるため, 参加者の発達段階や知識獲得段階は比較的近く, これらが近い学習者同士のペアで相互作用できたと言える.

これらの点を踏まえると, ペアプログラミングと協調学習には密接な関係があると言える. 今後は, ペアプログラミングを拡張していくことを考え, プログラミング学習を何人1グループで行うと効果的かという議論もでてくるであろう. グループサイズについて考察している中野の研究[中野, 2003]では, 話し合いの場において, 2人だと話がきちんと伝えあえ, 3人だと相乗効果が生まれやすく, 4名になるとアットホームでかつ多様性が生まれ,

5～6名を超えると関与度が低い者が出てくるということを指摘している。この指摘が、協調プログラミング学習にも当てはまるかどうかは今後検討していく必要がある。

(2) 学習者の振る舞い

2.1.4項で挙げた研究の中で作業中の学習者同士のインタラクションを分析している研究は、ペアプログラミングにおけるインタラクションを分析したものである。ペアプログラミングに限った話ではないが、参加する学習者の発達段階、知識獲得段階、コミュニケーションスキルによってインタラクションの内容や、その質が変わるため、学習者の振る舞いについて一般的に述べることは難しい。

その中で、Chenらは発話内容から Driver と Navigator には精神的な距離があること、作業中に何らかのコミュニケーション支援が必要であることを示した[Chen, 2007]。Chongらは、ペアが所有する専門的な知識が作業中のインタラクションに影響する可能性があること、キーボードの操作権 (Keyboard control) がペアの意思決定に影響する可能性があることを示した[Chong, 2007]。Bryantらは、作業中の発話プロトコル分析を行って Driver および Navigator の発話内容の分布を調査し、両者の分布に差はなく、同じレベルで相互作用していることを示した[Bryant, 2008]。本研究でも、作業中の学習者のインタラクションを分析し、会話と作業中に発生するつまずきの解決の成功・失敗との関係を示した。

対面環境における学習では、参加する学習者の発達段階、知識獲得段階、コミュニケーションスキルだけでなく、学習者の顔の表情やジェスチャなどのノンバーバルな情報も学習者の振る舞いに影響を与える可能性がある。今後も、それらを含めた分析やそれらを協調プログラミング学習支援にどのように生かしていくかという検討が必要となってくるであろう。

(3) 支援効果

アンケートの回答結果の分析のような定性的な分析においては、支援効果として次のようなものが挙げられている。

- プログラムのエラーの発見が容易になった
- お互いに教えあうことでプログラミングやフローチャートの理解が向上した
- 作成したフローチャートを評価することによって、プログラミングの良い具体例を見ることができた
- レビューの結果、フローチャートの誤りが発見できた

これらの結果は Peer Tutoring, Observation Learning の効果があったことを示したものである。つまり、Peer Tutoring, Observation Learning を取り入れた協調プログラミング学習が有効であることを示している。

定量的な分析では、プログラム作成課題を個人プログラミングとペアプログラミングで行わせたときに、後者の有効性について述べているものが多い。例えば、コードの質が高

かった [McDowell, 2002][Hanks, 2004] , 単位修得率が高かった [McDowell, 2002][Nagappan, 2003] , 中間および最終試験の成績が高かった [Nagappan, 2003] , 課題の提出率が上がった [Rountree, 2005] という結果が報告されている。また、ペアプログラミング実施前と実施後を比較したときについて、Urness らはコード理解やコード作成スキルが上がったと報告している [Urness, 2009]。本研究では、ペアプログラミング中の会話とプログラミング中に発生するつまずきの解決の成功・失敗との関係を示した。

いずれの研究でも協調的なプログラミング学習が有効であることを述べており、ピア・レビューやペアプログラミングのような学習方法を取り入れた協調プログラミング学習支援が有効であると考えられる。

5.2.3 今後の課題

本項では、本研究では明らかにできなかったが、今後、協調プログラミング学習支援について検討していくには重要であろうと考えられることを述べる。

本研究では、学習者が作成したプログラム（ソースコード）を分析することはしなかった。学習者が作成するソースコードは多種多様であり、それを評価指標として分析することは非常に困難であり、またそれを使って一般的な知見を出すことが難しいと考えたためである。本研究で実施した作問学習、ペアプログラミング学習では、どちらにおいても、多くとも 50 行程度のプログラムを扱った学習が行われた。例えば、Java のようなオブジェクト指向プログラミングでは、ソースコード行が長くなったり、複数のソースコードファイルを連携させたりすることがある。そのような言語のプログラミング学習では、本研究とは異なる知見が得られる可能性がある。

本研究では、分散環境での協調プログラミング学習における学習不振者について十分な考察をすることができなかった。これは、表 3-2 に示されているように、作問学習支援システムにおける掲示板投稿数が 2008 年度は 23 件、2009 年度は 9 件であるためであり、学習者同士のインタラクションを十分に分析できなかったことが原因である。しかし、別の見方をすれば、学習不振者について考えることも可能である。本研究で実施した作問学習支援システムで考えると、例えば、ある問題について評価が行われ、その評価に基づいて問題が修正された後に、問題の正確性評価等の評価値が上がった場合は学習がうまくいった、そうでない場合は学習がうまくいかなかったとすることで、学習不振者について考察することができる可能性がある。

4.5.1 項の発話長の分析において、失敗事例のほうが、成功事例よりも発話長が長いということが認められた。成功事例の平均発話長は 1.85 秒、失敗事例の平均発話長は 2.34 秒であり、その差は約 0.5 秒である。現実世界において、この差がどのような意味を示すのか、またこの差を考慮してどのように協調プログラミング学習を支援していくのかについては、今後考察していく必要がある。

本研究では、協調プログラミング学習支援について分析・考察を行ってきた。本研究で得られた知見は、例えば、グループで協調しながら数学の課題に取り組む等、他の協調作業へ応用することも可能であろう。本項で述べた課題も考慮しつつ、これから行われるであろう多くの協調プログラミング学習実践やそれに関連する実践について調査・分析を行うことで、協調プログラミング学習支援について、新たな知見を得ることができよう。

5.3 将来の展望

本研究で明らかにしたことを利用することで、学習に効果的な協調プログラミング学習支援環境を開発することができる可能性がある。分散環境における協調プログラミング学習の分析で得られた知見を利用して分散環境における学習支援環境を開発するということに限らず、例えば、分散環境における学習支援環境に対面環境で得られた知見を加味したり、対面環境における学習支援環境に分散環境で得られた知見を加味したりすることで、学習に効果的な支援環境を開発することができる可能性がある。本節では、本研究で得られた知見をもとに、今後の協調プログラミング学習支援の展望について述べる。

(1) 対面環境における協調プログラミング学習支援

本研究では、対面環境における協調プログラミング学習支援として、ペアプログラミング学習を実施した。本研究で設定したペアプログラミング学習に本研究で得られた知見を加味することで、学習に効果的な協調プログラミング学習ができる可能性がある。

図 5-1 に本研究で得られた結果をもとにした協調プログラミング学習支援案を示す。図 5-1 では、対面同期環境で実施するペアプログラミング学習を想定している。コンピュータには 2 台のマイクが接続されており、各々の学習者の発話を認識することができるものとする。

この環境において、次のプロセスでプログラミング学習することを考える。

1. 課題に対するプログラムを 2 人の学習者で相談しながら作成する
2. プログラム作成後、他学習者グループが作成したプログラムをレビューする
3. 1. で作成したプログラムを修正する

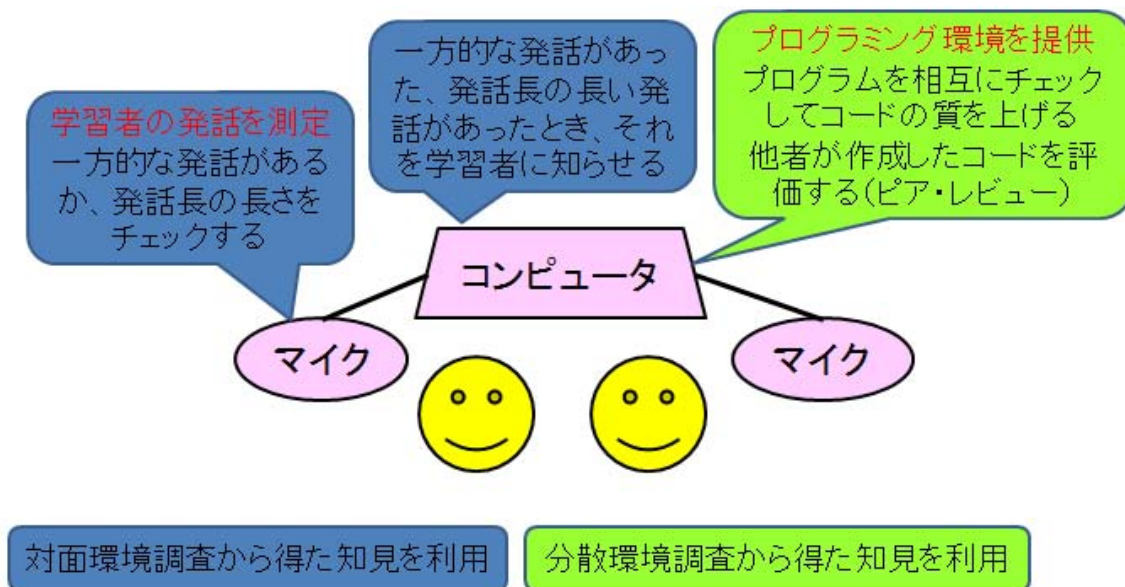


図 5-1 対面同期環境における協調プログラミング学習支援の構想

本研究で実施したペアプログラミング学習と違うところは、コンピュータが学習者の発話を認識することと、他のグループが作成したプログラムをレビューするという学習プロセスである。

プロセス 1 は、第 4 章で実施したペアプログラミング演習と同様のプログラミング手法であるが、ここでは、プログラミング中の 2 人の学習者の会話状況に応じてコンピュータが気づきを与える。学習者らが作成したプログラムをコンパイルしてエラーが発生した後、コンピュータは 2 人の学習者の会話において、一方的な発話があるかどうか、あるいは発話長が長い発話があるかどうかをチェックし、それらがあつた場合には学習者グループに知らせる。これにより、両者が対等にインタラクションできる機会を与える。プロセス 2 では、他学習者グループが作成したプログラムをレビューし、課題が完了していないグループは課題を完了させるためのアドバイスを受け、課題が完了しているグループは、課題が完了していないグループに対して課題を完了させるためのアドバイスをする。ペアプログラミングにピア・レビュー活動を加えることで、より深い理解ができる可能性がある。これにより、学習不振者を少なくすることができる可能性もある。プロセス 3 では、プロセス 2 で受けたアドバイスをもとに、プロセス 1 で受けたプログラムを修正する。

このような支援環境を開発する場合は、例えば、プロセス 1 においてどのような基準で学習者に気づきを与えるのか、どのタイミングでプロセス 1 からプロセス 2 へ移行するのかについて検討していく必要はある。

(2) 分散同期環境における協調プログラミング学習支援

本研究では、分散環境における協調プログラミング学習の実践として、作問学習支援シ

システムを分散非同期環境で利用した。5.2.3 項でも述べたが、本研究では作問学習支援システムにおいて、学習者同士のインタラクションが密に行われたとは言えなかった。これは、分散非同期環境で利用したということも考えられる。そこで、作問学習を利用した協調プログラミング学習を対面同期環境で実施することにより、学習者同士のインタラクションを密に行うことができるようになる可能性がある。

図 5-2 は分散同期環境において、協調プログラミング学習を実施することを想定した図である。コンピュータにはマイクが接続されており、学習者の発話が認識できるものとする。このような環境で協調的にプログラミング学習を行い、さらに、コンピュータは 2 人の学習者の会話において、一方的な発話があるかどうか、あるいは発話長が長い発話があるかどうかをチェックし、それらがあった場合には学習者グループに知らせることで、学習に効果的な協調プログラミング学習ができる可能性がある。

分散非同期環境では、学習者同士のインタラクションを密に行うことは難しいかもしれないが、図 5-2 のような設定で協調プログラミング学習をすることにより、分散環境における協調プログラミング学習中の学習者同士のインタラクションについても考察することができるようになる。

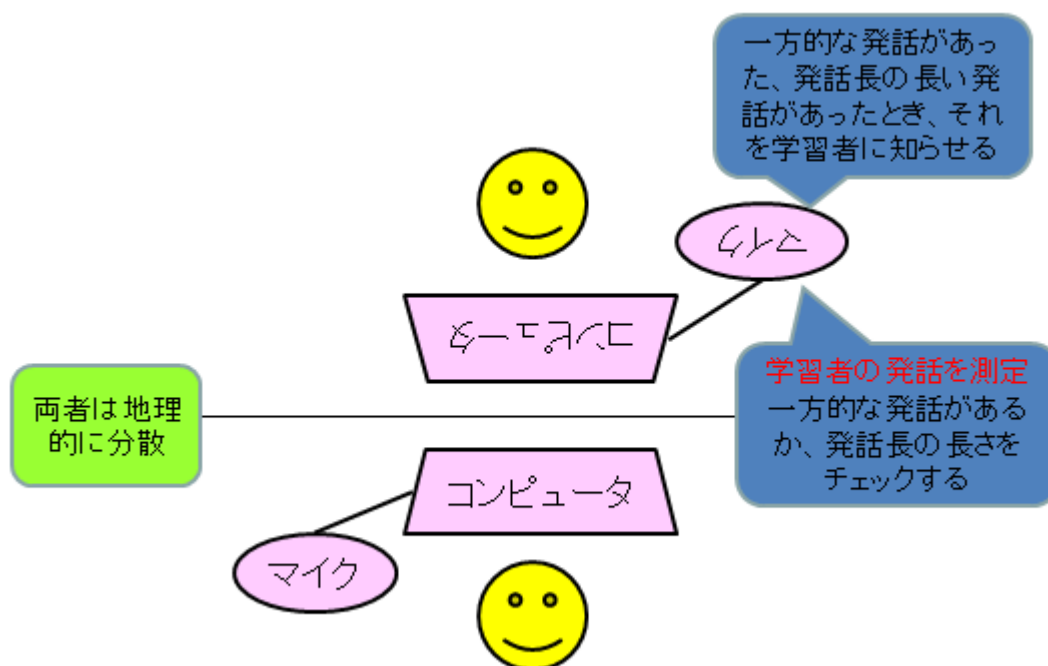


図 5-2 分散同期環境における協調プログラミング学習支援の構想

(3) 分散非同期環境における協調プログラミング学習支援

本研究で実施した分散非同期環境における協調プログラミング学習でも分かったことだが、学習支援システム上で学習者同士のインタラクションが密に行われたとは言えなかった。この結果は、分散非同期環境下で協調的にプログラミング学習を行うことは難しいこ

とを示唆している可能性もある。協調的にプログラミング学習を行うための工夫について考察するには、実践例を増やすなどして、さらなる調査が必要であると考えられる。例えば、学習者同士でインタラクションをさせるためのモチベーションをシステムが与えたり、グループ内の他学習者が意見を求めていることをシステムが知らせたりすることで、学習者同士のインタラクションを密にさせることは可能かもしれない。

学習者同士のインタラクションについて考えていくことは必要であると考えられるが、一方で、既存研究で実施された Wiki やデータベースを利用した学習や、本研究で実施した作問学習では、協調プログラミング学習が有効であったことが示されているので、現実的にはこれらのような学習が分散非同期環境で行う協調プログラミング学習において適切であるかもしれない。

本節では、対面環境における協調プログラミング支援、分散環境における協調プログラミング学習支援について展望を述べた。既存研究や本研究では、それぞれの環境で様々な支援効果が得られている。これらの支援効果は環境特有の効果もあれば、そうでない効果もある。また、情報技術の急速な発達につれて、今後は、分散同期環境で実施するビデオ会議のような、分散環境にありながら対面環境と同等の状況を実現した場面における協調プログラミング学習も行われる可能性がある。今後も分散・対面の両環境あるいは新たな環境における協調プログラミング学習支援の支援効果に注目し、支援環境の開発に生かしていきたい。

第6章 結論

今日までに提案されている学習支援方法は多種多様化しており、また、今後も支援対象となる学習形態が拡大していく可能性がある。協調プログラミング学習支援においても、ピア・レビューを導入する手法、グループ活動・他者活動などを取り入れた授業展開を行う手法、話し合い活動のような共同学習を取り入れる手法があり、現在も支援対象の拡大が進んでいる。学習支援システムを開発するためには学習者の学習形態、学習状態、学習目的を考慮する必要がある。システムを開発するためには、それとは別に、既存の支援環境において得られた知見を調査・分析する必要がある。

本研究では、今後の協調プログラミング学習支援のために、既存の支援において、得られた知見を調査しまとめた。特に分散環境における支援と対面環境における支援という観点でまとめた。その結果、分散環境における協調プログラミング学習支援例が少ないこと、協調プログラミング学習中における学習不振者に関する考察が十分に行われていないことの2つの問題点を指摘し、問題の解決を図った。

問題解決のため、本研究では、分散環境における協調プログラミング学習と対面環境における協調プログラミングを実施し、それぞれの学習において、学習者同士のやりとりや学習者の成績について調査・分析した。

分散環境における協調プログラミング学習として、分散非同期環境で利用する作問学習支援システムにおいて実施したプログラミング学習を調査・分析した。学習者のシステム利用状況と学習者の成績の関係について分析した結果、次の3点が分かった。

- システムを積極的に利用すること、学習者が作成した問題に多く解答すること、学習者が作成する問題の正確性が高いことが、成績向上に関係している
- 問題に解答することが成績向上に大きく関係している
- 正確性の低い問題を作成することが成績低下に大きく関係している

また、対面環境における協調プログラミング学習として、ペアプログラミング学習を分析した。プログラミング中に発生するつまずきの解決に成功・失敗とプログラミング中の会話の関係について分析した結果、失敗した事例について、成功した事例と比較したときに、次の3点が分かった。

- 1回の発話長が長い
- 説明を繰り返す回数が多い
- 一方的に発話する回数が多い

これらの結果を踏まえ、改めて、協調プログラミング学習支援のために、既存の支援において、学習者がどのような振舞いをしたのか、学習者が得られた効果は何なのかという

こと等を考察した。5.3節で述べたように、本研究で明らかにしたことを生かし、協調プログラミング学習を支援するための環境を開発することはできる。それが学習者の効果的な学習につなげることができるかどうかは本研究の結果からだけではわからない。今後は協調プログラミング学習支援環境を開発し、学習者が効果的なプログラミング学習をすることができるかどうかについて調査していくことが必要となるだろう。それを行うことで、新たな学習形態における実践例を示すことができ、それを分析することで、その後の支援に生かすことも可能となる。開発と分析を繰り返していくことで、よりよい協調プログラミング学習支援が行われることを期待したい。

謝辞

本論文をまとめるにあたり，副指導教員である筑波大学図書館情報メディア系所属の井上智雄准教授には終始多大なご助言を頂きました。2009年の博士後期課程入学前から，研究の進め方，論文の書き方，発表の仕方等，多岐に渡るご指導を頂きました。また，ティーチング・アシスタント，リサーチ・アシスタント，日本学術振興会特別研究員に就く機会を与えて頂いたことにより，研究や学生指導に関する知識と経験を得ることができました。

主指導教員である筑波大学図書館情報メディア系所属の杉本重雄教授，副指導教員である筑波大学図書館情報メディア系所属の佐藤哲司教授，筑波大学図書館情報メディア系所属の田中和世教授には，論文審査で貴重なご助言を頂くとともに激励の言葉を頂きました。

本論文の審査にあたり，杉本重雄教授，井上智雄准教授，佐藤哲司教授，田中和世教授とともに論文審査委員を務めていただいた東京学芸大学大学院連合学校教育学研究科所属の宮寺庸造教授には，論文審査で貴重なご助言を頂くとともに，博士論文発表に関しても貴重なご意見を頂きました。

著者の学部生時代，修士時代の主指導教員である東京学芸大学大学院連合学校教育学研究科所属の樋山淳雄教授には，著者の修士課程修了後も研究や論文執筆に関して多大なご助言，ご助力を頂きました。また，博士論文発表に関しても貴重なご意見を頂き，博士論文最終発表会でも貴重なご意見を頂きました。

著者は2011年4月より日本学術振興会特別研究員(DC2)に着任し，日本学術振興会には，科学研究費補助金特別研究員奨励費（課題番号 23・2956：遠隔および対面インタラクショナルマイニングに基づく協調学習支援システムの開発）等，本研究の一部を補助して頂きました。

著者が発表した研究論文の共著者である高川祐輔氏，研究協力者である鄭乃文氏および任海因氏，英文執筆の協力者である Nawahdah Mamoun 氏をはじめとして，筑波大学図書館情報メディア系の井上研究室の皆様には，日常生活や研究室活動において，多大なご支援を頂きました。

著者らの家族，筑波大学図書館情報等支援室に所属する職員の方々，研究会や国際会議等の学外発表においてご意見を頂いた方々，論文投稿にあたり貴重な査読コメントを頂いた方々の支援もあり，これまでの研究成果を出すに至りました。

以上，すべての方々に，厚く御礼を申し上げます。

参考文献

[Agraval, 1993] Agraval, R., Imielinski, T., and Swami, A. N.: Mining association rules between sets of items in large database, Proc. of SIGMOD, pp. 207-216 (1993)

[Agraval, 1994] Agrawal, R., and Srikant, R.: Fast Algorithms for Mining Association Rules, Proc. of VLDB, pp.487-499 (1994)

[Bandura, 1971] Bandura, A.: Social Learning Theory. New York: General Learning Press (1971)

[Beck, 1999] Beck, K.: Extreme Programming Explained: Embrace Change, Reading, PA: Addison-Wesley (1999)

[Bryant, 2008] Bryant, S., Romero, P., and Boulay, B.: Pair Programming and the Mysterious Role of the Navigator, International Journal of Human-Computing Study, Vol. 66, No. 7, pp. 519-529 (2008)

[Chen, 2007] Chen, W. and Nordbo, M.: Understanding Pair-Programming from a Socio-cultural Perspective, Proc. of Computer Supported Collaborative Learning (CSCL), pp. 138-140 (2007)

[Chong, 2007] Chong, J., and Hurlbutt, T.: The Social Dynamics of Pair Programming, Proceedings of the 29th International Conference on Software Engineering (ICSE), pp. 354-363 (2007)

[Constantino-Gonzalez, 2000] Constantino-Gonzalez, M. A., and Suthers, D. D.: A Coached Collaborative Learning Environment for Entity-Relationship Modeling, Proc. of ITS, pp.544-553 (2000)

[Denny, 2008] Denny, P., Hamer, J., L-Reily, A., and Purchase, H.: PeerWise: Students Sharing their Multiple Choice Question, Proc. of ICER'08 (2008)

[Hamalainen, 2006] Hamalainen, W., Laine, T. H., and Sutinen, E.: Data mining in Personalizing Distance Education Courses, Data Mining in e-Learning (Eds: Romero, C. and Ventura, S.), WIT Press, pp.157-171 (2006)

[Hanks, 2004] Hanks, B., McDowell, C., Draper, D., and Krnjajic, M.: Program Quality with Pair Programming in CS1, Proc. of Innovation and Technology in Computer Science Education (ITiCSE), pp.176-180 (2004)

[Hayashi, 2008] Hayashi, T., Mizuno, T., Tominaga, H., Tarumi, H., and Yamasaki, T.: A Learner-centered e-Learning Environment by Contributing and Sharing Problems Created by Learners, Proc. of CollabTech'08, pp. 94-98 (2008)

[Heh, 1996] Heh, J., Shu, W., Jehng, J. J., and Chan, T.: Design and Development of a Distributed Multi-User Visual Learning Environment, Proc. of ITS, pp.187-196 (1996)

[Hirai, 2007] Hirai, Y., and Hazeyama, A.: A Learning Support System based on Question-posing and Its Evaluation, Proc. of C5'07, IEEE Computer Society Press, pp. 180-185 (2007)

[Hirai, 2009] Hirai, Y., Takagawa, Y., Hazeyama, A., and Inoue, T.: Analysis of the Learners' Assessment Activity in a Collaborative Learning Support System Based on Question-posing, Proc. of ICCE'09 workshop, pp.25-29 (2009)

[Hirai, 2010] Hirai, Y., Inoue, T.: Analysis of Learners' Activity in a Question-Posing Learning Support System by Association Rule Mining, Proceedings of the 4th International Workshop of Modeling, Management and Generation of Problems/Questions in Technology-Enhanced Learning (ICCE2010WS), Putrajaya, Malaysia (2010)

[Hirashima, 2008] Hirashima, T., Yokoyama, T., Okamoto, M., et al.: Long-term Use of Learning Environment for Problem-Posing in Arithmetical Word Problems, Proc. of ICCE, pp. 817-824 (2008)

[McDowell, 2002] McDowell, C., Werner, L., Bullock, H., and Fernald J.: The Effects of Pair-Programming on Performance in an Introductory Programming Course, Proc. of ACM SIGCSE, pp. 38-42 (2002)

[Nagappan, 2003] Nagappan, N., Williams, L., Ferzli, M., Wieve, E., Yang, K., Miller, C., and Balik, S.: Improving the CS1 Experience with Pair Programming, Proc. of ACM SIGCSE, pp. 359-362 (2003)

[Nakano, 1999] Nakano, A., Hirashima, T., Takeuchi, A.: Problem-Making Practice to Master Solution-Methods in Intelligent Learning Environment, Proc. of ICCE'99, pp. 891-898 (1999)

[Preston, 2006] Preston, D.: Using Collaborative Learning Research to Enhance Pair Programming Pedagogy, ACM SIGITE Newsletter, Vol.3, No.1, pp. 16-21 (2006)

[Rountree, 2005] Rountree, J., Rountree, N., Robins, A., and Hannah, R.: Observations of Student Competency in a CS1 Course, Proc. of Australasian Computing Education Conference, pp. 145-149 (2005)

[Terada, 2005] Terada, M.: ETV: a program trace player for students, Proc. ITiCSE'05, Vol.37, No.3, pp.118-122 (2005)

[Urness, 2009] Urness T.: Assessment Using Peer Evaluations, Random Pair Assignment, and Collaborative Programming in CS1, Proc. of the Consortium for Computing Sciences in Colleges, pp. 87-93 (2009)

[Wing, 1998] Wing, L. K., and Warren, J.: Reusing of Teaching Materials for an Intelligent Tutoring System with Multiple Teaching Strategies, Proc. of ICCE, Vol.1, pp.536-543 (1998)

[Wray, 2010] Wray, S.: How Pair Programming Really Works, IEEE Software, Jan./Feb. 2010, pp. 50-55 (2010)

[Yu, 2005] Yu, F.-Y., Liu, Y.-H. and Chan, T.-W.: A web-based learning system for question posing and peer assessment, Innovations in Education and Teaching International, Vol. 42, No. 4, pp. 337-348 (2005)

[Yu, 2008] Yu, F.-Y., and Liu, Y.-H.: Creating a psychologically safe online space for a student-generated questions learning activity via different identity revelation modes,

British Journal of Educational Technology (2008)

[arules] arules: Mining Association Rules and Frequent Itemsets, The Comprehensive R Archive Network, <http://cran.r-project.org/web/packages/arules>

[iCorpusStudio] iCorpusStudio: <http://www.ii.ist.i.kyoto-u.ac.jp/iCorpusStudio/>

[R Project] The R Project for Statistical Computing, <http://www.r-project.org/>

[Zusaku] <http://www.zusaku.com/>

[赤堀, 2002] 赤堀侃司: 教育工学への招待: 教育の問題解決の方法論, ジャストシステム (2002)

[阿部, 2009] 阿部圭一: C 言語によるプログラミング教育についての省察, 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.2009, No. 15, pp.205-212 (2009)

[伊藤, 2000] 伊藤良二, 小西達裕, 伊東幸宏: プログラムの問題領域上での動作説明を行うプログラミング学習支援システムの構築, 人工知能学会論文誌, Vol.15, No.2, pp.362-375 (2000)

[飯田, 2008] 飯田周作, 飯田千代, 清藤武暢, 佐藤創: アルゴリズム的思考法の教育, 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.57-64 (2008)

[稲葉, 1999] 稲葉晶子, 豊田順一: CSCL の背景と研究動向, 教育システム情報学会誌, Vol.16, No.3, pp.111-120 (1999)

[今泉, 2010] 今泉俊幸, 橋浦弘明, 松浦佐江子, 古宮誠一: ブロック構造の可視化によるプログラミング学習支援環境 azur～関数の動作の可視化～, 情報処理学会研究報告, ソフトウェア工学研究会報告, Vol. 2010, No. 11, pp.1-7 (2010)

[植野, 2007] 植野真臣: eラーニングにおけるデータマイニング, 日本教育工学会誌, Vol. 31, No. 3, pp. 271-283 (2007)

[内田, 2007] 内田保雄: 初級プログラミング技法学習のための自動問システム, 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. 2007, No. 123, pp.109-113 (2007)

[岡本, 1996] 岡本真彦: 問題解決スキーマの獲得における問題作成の効果, 日本教育心理学会第 38 回大会発表論文集, Vol. 38, p. 368 (1996)

[大槻, 1993] 大槻説平: 発見的学習とその支援環境, 人工知能学会誌, Vol.8, No.4, pp.411-418 (1993)

[大西, 2010] 大西建輔: プログラミング, 何をどう教えているか, 連載開始にあたって, 情報処理, Vol. 51, No. 10, p. 1341 (2010)

[梶田, 2009] 梶田建夫, 程鵬: アルゴリズム/プログラミング学習支援システムの開発, 名古屋学院大学論集, 人文・自然科学篇, Vol.48, No.2, pp.11-27 (2009)

[教育工学, 2000] 日本教育工学会編: 教育工学事典, 実教出版 (2000)

[喜多, 2005] 喜多義弘, 川添貴議, 片山徹郎: Java プログラム自動可視化ツール Avis におけるクラス構造可視化のための拡張, 電子情報通信学会技術研究報告, Vol.105, No.490, pp.7-12 (2005)

[北, 2007] 北栄輔, 山梨樹里: Peer Review に基づいたプログラミング実習授業支援ツールの開発, 名古屋高等教育研究, No.7, pp.341-353 (2007)

[倉田, 2007] 倉田英和, 富永浩之, 林敏浩, 垂水浩幸: 実行テストによるプログラム判定を用いた初級 C プログラミング演習支援と授業実践, 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. 2007, No. 11, pp.11-18 (2007)

[小尻, 2003] 小尻智子: 学習状況把握機構を備えた対面的協調学習支援システムに関する研究, 名古屋大学博士学位論文 (2003)

[佐合, 2000] 佐合尚子, 竹田尚彦: RPGによりコミュニケーション能力を高める英会話 CAL, 情報処理学会研究報告, Vol.2000, No.117, pp.13-20 (2002)

[匂坂, 2009] 匂坂智子, 渡辺成良: プログラミング初学者の学習方略と段階的理解度に関する調査および支援ルールの作成について, 教育システム情報学会誌, Vol. 26, No. 1, pp. 5-15 (2009)

[新開, 2008] 新開純子, 炭谷真也: プロセスを重視したプログラミング教育支援システムの開発, 日本教育工学会論文誌, Vol.31, pp.45-48 (2008)

[新開, 2009] 新開純子, 宮地功: プロセスの重視と評価活動を取り入れた C プログラミング教育の効果, 教育システム情報学会誌, Vol. 26, No. 1, pp. 16-25 (2009)

[菅原, 2007] 菅原典子, 織田恵太, 赤池英夫, 角田博保: 集合教育に用いる即応型 e-ラーニングシステム SHoes における組織学習支援, 情報処理学会論文誌, Vol. 48, No. 8, pp. 2791-2801 (2007)

[スリーマン, 1987] スリーマン, D, ブラウン, J.S (編): 人工知能と知的 CAI システム, 講談社 (1987)

[先進学習基盤協議会, 2001] 先進学習基盤協議会(編): e ラーニング白書 2001/2002 年版, オーム社開発局 (2001)

[高木, 2007] 高木正則, 田中充, 勅使河原可海: 学生による問題作成およびその相互評価を可能とする協調学習型 WBT システム, 情報処理学会論文誌, Vol. 48, No. 3, pp. 1532-1545 (2007)

[高橋, 2007] 高橋麻奈: やさしい C 第 3 版, ソフトバンク (2007)

[瀧口, 2002] オブジェクト指向を学習するためのゲーム型 CAL の開発, 電子情報通信学会技術研究報告, Vol.12, No.330, pp.1-4 (2002)

[谷口, 2005] 谷口るり子: Wiki を利用した協調的プログラミング学習, 論文誌 IT 活用教育方法研究, Vol.8, No.1, pp.56-60 (2005)

[東本, 2007] 東本崇仁, 市将治, 平嶋宗, 竹内章: 多桁減算を対象とした作問学習支援環境の設計・開発, 日本教育工学会論文誌, Vol. 31, No. 1, pp. 61-68 (2007)

[富永, 2008] 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. 42, pp. 49-56 (2008)

[中尾, 2006] 中尾茂子, 安達一寿: e-learning 学習支援システムの協調学習ツールを利用し

たプログラミング演習の実践と評価, 日本教育工学会論文誌, Vol.30(Suppl.), pp.145-148 (2006)

[中野, 2000] 中野明, 平嶋宗, 竹内章: 「問題を作ることによる学習」の知的支援環境, 電子情報通信学会論文誌 D-I, Vol. J83-D-I, No. 6, pp. 539-549 (2000)

[中野, 2002] 中野明, 平嶋宗, 竹内章: 演算の理解を指向した知的作問学習支援環境, 人工知能学会論文誌, Vol. 17, No. 5, pp. 598-607 (2002)

[中野, 2003] 中野民夫: ファシリテーション革命: 参加型の場づくりの技法, 岩波書店 (2003)

[仲林, 1997] 仲林清, 小池義昌, 丸山美奈, 東平洋史, 福原美三, 中村行宏: WWW を用いた知的 CAI システム CALAT, 電子情報通信学会論文誌 D-II, Vol.J80-D-II, No.4, pp.906-914 (1997)

[生田目, 2004] 生田目康子: ピア・レビューをともなうグループ学習の評価——斉型プログラミング授業への適用——, 情報処理学会論文誌, Vol. 45, No. 9, pp. 2226-2235 (2004)

[農林水産省, 1996] 農林水産省農林水産技術会議事務局 (編): 農林水産業の高度情報システム, 農林水産省 (1996)

[浜野, 2007] 浜野康裕, 天寄聡介, 水野修, 菊野亨: 相関ルールマイニングによるソフトウェア開発プロジェクト中のリスク要因の分析, コンピュータソフトウェア, Vol. 24, No. 2, pp.79-87, 日本ソフトウェア科学会 (2007)

[林, 2004] 林敏浩: 教育工学研究: IT 活用による新しい教育・学習環境を目指して, 香川大学総合情報センター年報, No.2, pp.5-9 (2005)

[平井, 2008] 平井佑樹, 樫山淳雄: 作問に基づく協調学習支援システムとその分散非同期学習環境への適用, 情報処理学会論文誌, Vol. 49, No. 10, pp. 3341-3353 (2008)

[平井, 2010] 平井佑樹, 樫山淳雄, 井上智雄: 学習者による作問に基づく学習支援システムの分散非同期環境への適用とその効果, 教育システム情報学会論文誌, Vol. 27, No. 1, pp.62-73 (2010)

[平井, 2012] 平井佑樹, 井上智雄: ペアプログラミング学習における状態の推定 -つまずきの解決の成功と失敗に見られる会話の違い-, 情報処理学会論文誌, Vol. 53, No. 1 (2012)

[平嶋, 2001] 平嶋宗, 梅田 多一, 志岐 隆弘, 竹内 章: XML を用いた算数の文章問題の作成・共有環境一つのかめ算等の特定の計算手順で解決可能な文章問題を対象として, 教育システム情報学会誌, Vol. 18, No. 3, pp. 284-296 (2001)

[平嶋, 2005a] 平嶋宗: 「問題を作ることによる学習」の分類と知的支援の方法, 教育システム情報学会研究報告, Vol.20 No.3, pp.3-10 (2005).

[平嶋, 2005b] 平嶋宗: 作問学習のモデル化, 教育システム情報学会全国大会, Vol. 30, pp.37-38 (2005)

[堀口, 2001] 堀口知也, 平嶋宗: 誤りへの気づきを支援するシミュレーション環境・表現方法と視覚効果を考慮した Error-Based Simulation の制御-, 教育システム情報学会誌, Vol.18, No.3,4, pp.364-376 (2001)

[松田, 1997] 松田憲幸, 柏原昭博, 平嶋宗, 豊田順一: プログラムの振舞いに基づく再帰プログラミングの教育支援, 電子情報通信学会論文誌, Vol.80, No.1, pp.326-335 (1997)

[溝口, 1995] 溝口理一郎: 知的教育システム, 情報処理学会論文誌, Vol.36, No.2, pp.135-140 (1995)

[皆月, 2009] 皆月昭則, 林秀彦: 失敗ソースコードを用いたプログラミング授業環境に関する研究-情報系を専攻としないしない学生による参画型授業モデルの考察-, 鳴門教育大学情報教育ジャーナル, No.6, pp.37-43 (2009)

[山田, 2010] 山田雅之: オンデマンド講義と Jigsaw 形式による協調学習のブレンデッドの効果, 教育システム情報学会誌, Vol. 27, No. 1, pp.14-20 (2010)

[山本, 2006] 山本利一, 田賀秀子, 新屋智絵, 小林靖英: 共同学習を取り入れたプログラミング学習の課題の提案-カーリングゲームを取り入れたプログラミング指導-, 日本教育情報学会誌「教育情報研究」, Vol.22, No.3, pp.11-18 (2006)

[横山, 2005] 横山琢郎, 平嶋宗, 岡本真彦, 竹内章: 統合レベルでの作問を支援する学習環境の設計・開発と小学校低学年での学習効果, 人工知能学会第 19 回全国大会講演論文,

2A1-03 (2005)

[横山, 2007] 横山琢郎, 平嶋宗, 岡本真彦, 竹内章: 単文統合による作問を対象とした学習支援システムの長期的利用とその効果, 日本教育工学会論文誌, Vol. 30, No. 4, pp. 333-341 (2007)

[黎, 1998] 黎翔寧, 宮寺庸造, 横山節雄, 樫山淳雄: グループプログラミング教育システムの開発, 電子情報通信学会技術報告, ET98-103 (1998)

[渡辺, 2005] 渡辺博芳: 情報基礎教育における「ディスカッション」を用いた協同学習活動, 第3回日本 WebCT ユーザカンファレンス予稿集, pp. 83-88 (2005)

研究業績

学術雑誌掲載論文

- [1] 平井佑樹, 井上智雄 : ペアプログラミング学習における状態の推定—つまずきの解決の成功と失敗に見られる会話の違い, 情報処理学会論文誌, Vol. 53, No. 1, pp.72-80 (2012)
- [2] 平井佑樹, 樫山淳雄, 井上智雄 : 学習者による作問に基づく学習支援システムの分散非同期環境への適用とその効果, 教育システム情報学会論文誌, Vol. 27, No. 1, pp.62-73 (2010)
- [3] 平井佑樹, 樫山淳雄 : 作問に基づく協調学習支援システムとその分散非同期学習環境への適用, 情報処理学会論文誌, Vol. 49, No. 10, pp. 3341-3353 (2008)

査読付き国際会議論文

- [1] Yuki Hirai, Tomoo Inoue: Toward better collaborative problem-solving in programming learning: use of pair programming and its observation, Workshop Proceedings of the 19th International Conference on Computers in Education (ICCE2011), pp.362-369, Chiang Mai, Thailand (2011)
- [2] Yuki Hirai, Tomoo Inoue: Analysis of Learners' Activity in a Question-Posing Learning Support System by Association Rule Mining, Workshop Proceedings of the 18th International Conference on Computers in Education (ICCE2010), pp.70-78, Putrajaya, Malaysia (2010)
- [3] Yuuki Hirai, Yusuke Takagawa, Atsuo Hazeyama and Tomoo Inoue: Analysis of the Learners' Assessment Activity in a Collaborative Learning Support System Based on Question-posing, Proceedings of the 3rd International Workshop of Modeling, Management and Generation of Problems/Questions in Technology-Enhanced Learning (ICCE2009WS), pp. 25-29, HongKong (2009)
- [4] Yuuki Hirai, Atsuo Hazeyama, Tomoo Inoue, Assessment of Learning in Concerto III: A Collaborative Learning Support System Based on Question-posing,

Proceedings of the 12th IASTED International Conference on Computers and Advanced Technology in Education (CATE2009), pp. 36-43, St. Thomas, US Virgin Islands (2009)

- [5] Yuuki Hirai, Atsuo Hazeyama : Development and Application of Concerto II : Question-posing Based Collaborative Learning Support System, Proc. of CollabTech'08, pp. 74-78, Wakayama, Japan (2008)
- [6] Atsuo Hazeyama and Yuuki Hirai : Concerto II : A Learning Community Support System Based on Question-posing, Proc. of ICALT'07, pp. 338-339, Niigata, Japan (2007)
- [7] Yuuki Hirai and Atsuo Hazeyama : A Learning Support System based on Question-posing and Its Evaluation, Proc. of C5'07, IEEE Computer Society Press, pp. 180-185, Kyoto, Japan (2007)

著書

- [1] Atsuo Hazeyama and Yuuki Hirai: Concerto II: A Collaborative Learning Support System Based on Question Posing, Advanced Learning, Chapter 3, In-Teh Press (2009)

講演

- [1] 平井佑樹 : 作問に基づく協調学習支援システムとその評価, 創価大学2008年度現代GPフォーラム, 関連事例報告 (2009)

その他の論文・発表

- [1] 平井佑樹, 井上智雄 : ペアプログラミング学習における会話と問題解決の関係, 電子情報通信学会研究報告, Vol.111, No.190, pp.61-66, (ヒューマンコミュニケーション基礎 HCS2011-38 (2011-08)) (2011)
- [2] 平井佑樹, 井上智雄 : 作問学習支援システムにおける学習者の行動と成績との関係, 人工知能学会第59回先進的学習科学と工学研究会資料, SIG-ALST-B001-08 (2010)
- [3] 平井佑樹, 樫山淳雄 : 学習者による作問に基づく協調学習支援システムの大学の講義への適用効果, 情報処理学会コンピュータと教育研究会第98回研究会 (2009)

- [4] 平井佑樹, 富田高史, 樫山淳雄 : 学習内容理解の促進に注目した作問学習支援システムの提案, 情報処理学会第70回全国大会 (2008)

- [5] 林赤団, 平井佑樹, 樫山淳雄 : グループによるソフトウェア開発演習におけるマインドマップを応用した知識共有支援システム, 情報処理学会第70回全国大会 (2008)

- [6] 平井佑樹, 樫山淳雄 : 作問に基づく学習コミュニティ支援システムとその適用, 教育システム情報学会2007年度第3回研究会, pp. 6-13 (2007)

- [7] 平井佑樹, 樫山淳雄 : Concerto II : 作問に基づく学習コミュニティ支援システム, 教育システム情報学会第32回全国大会講演論文集, pp. 446-447 (2007)

- [8] 平井佑樹, 樫山淳雄 : 作問による学習支援システムとその評価, 情報処理学会グループウェアとネットワークサービス研究会ワークショップ2006講演論文集, pp. 7-12 (2006)

- [9] 平井佑樹, 樫山淳雄 : 作問による協調学習支援システムの構築, 情報処理学会第68回全国大会講演論文集, No. 4, pp. 457-458 (2006)