

2022年度 博士論文

2022 Dissertation

情報構造指向アプローチに基づく知的学習支援システムの
汎化を指向した多角的発展に関する研究

Multidirectional Development for Generalization of Intelligent Tutoring System

Based on Information-structure-oriented Approach

古池 謙人

Kento Koike

2023年2月1日

東京工芸大学大学院工学研究科 電子情報工学専攻

本論文は東京工芸大学大学院工学研究科に
博士（工学）授与の要件として提出した博士論文である。

古池 謙人

審査委員：

主査	東本 崇仁 准教授
副査	上倉 一人 教授
副査	片上 大輔 教授
副査	姜 有宣 教授
外部副査	平嶋 宗 教授（広島大学）
外部副査	赤倉 貴子 教授（東京理科大学）

情報構造指向アプローチに基づく知的学習支援システムの 汎化を指向した多角的発展に関する研究*

古池 謙人

内容梗概

近年、集団教育が主流化する教授者不足のなかで適応的な個別指導を実現する一つのアプローチが約半世紀の間に発展してきた。それは、計算機による知的な学習環境、すなわち知的学習支援システム (Intelligent Tutoring System: ITS) の実現である。

ITS において、学習活動の前提となるモデルを記述することは知的な振舞いを実現するうえで重要である。一般に ITS の主要な構成要素を担うモデルとしては、領域知識モデル、学習者モデル、教授モデルが挙げられる。特に、領域知識モデルは学習者の状態表現や教授内容の根幹を担うことから、ITS にとって必要不可欠な要素である。この領域知識モデルを設計する手法の一つに、情報構造指向アプローチがある。情報構造指向アプローチとは、システムが学習課題の文脈に沿った診断・推論・フィードバックを実施するために、学習課題に用いられる領域知識を計算機可読な状態で記述するモデル構築手法である。特に本論文では、このアプローチのうち領域知識を構成要素に分節化し、その集合として規定するモデルを設計する方法を対象とする。

情報構造指向アプローチに基づいて開発された ITS では、対象領域に特化した領域知識モデルが活用される。これにより、ITS は対象領域の知識や特有の誤りに踏み込んだ適応的支援を網羅的に設計することが可能になる。一方、このような ITS においては、領域知識モデルを対象領域に特化して作り込む必要があることから、対象領域の緻密な分析が必須となる。ゆえに、何を学ぶべきかが自明ではなく解も一意に規定しえないような対象領域に対しての開発が困難である。それに加えて、領域知識モデルを独立して再利用や拡張をするための知見が少ないことが、ITS1 つあたりの開発コストを高めている。情報構造指向アプローチに基づいて作られた領域知識モデルの独立性やモジュール性など、その汎用性を示唆する指摘もなされているが、実際に当該のモデルを再利用ないし他の教授モデルに適用するなどといった情報構造指向アプローチの汎用性を示した研究は見当たらない。よって、情報構造指向アプローチに基づく ITS の開発が汎化されているとは言いづ

*東京工芸大学大学院 工学研究科 電子情報工学専攻, 2022 年度 博士論文, 2023 年 2 月 1 日。

らいのが現状である。

そこで、本論文では情報構造指向アプローチに基づく ITS の汎化を目的として、以下に述べる 3 つの課題を設定し、多角的な発展を試みた。それは、(1) 未知の対象領域に対する領域知識モデル構築法の実現、(2) 領域知識モデルの汎用的な再利用法の実現、(3) 領域知識モデルの高次化を伴う汎用的な拡張法の実現、である。

本論文の構成は次のとおりである。第 I 部では、上記に示した内容を 1 章にて序論、2 章にて本研究の目的とアプローチとして述べた。第 II 部では、1 つ目の課題に対し、従来研究の対象領域と比較してより未知な対象領域、すなわち未成熟かつ複雑な対象領域における領域知識モデル構築法に関する知見を創出することを目的とした研究について述べた。具体的には、プログラミング領域を対象とし、3 章で新たな教授モデルの研究、4 章で問題解決過程モデルおよびそれに付随する領域知識モデルの研究、5 章で学習者モデルの研究について述べた。これにより、情報構造指向アプローチに基づいた領域知識モデルをプログラミング領域で構築しうることを、一連の研究を通して確認し、未知領域において領域知識モデルを構築する方法の知見を示した。第 III 部では、2 つ目の課題に対し、既存の領域知識モデルを再利用可能かつドメイン独立な教授モデルを実現し、モデルの再利用法および再利用可能なモデルについての知見を得ることを目的とした研究について述べた。具体的には、6 章にて Learning-by-teaching を対象とした領域独立な教授モデルの研究、7 章にてその教授モデルに対する感情的側面の支援可能性を検証する研究について述べた。よって、情報構造指向アプローチに基づく既存領域知識モデルを、感情的支援を含むかたちで他の教授モデルに再利用する方法に関する知見を一連の研究から示した。第 IV 部では、3 つ目の課題に対し、既存の領域知識モデルにおける学習対象の高次化を伴う拡張法に対する知見の創出を目的とした研究について述べた。具体的には、8 章にて学習者のメタ認知活動を促すメタ認知駆動学習を支援するための領域知識を既存領域知識モデルに拡張する研究について述べた。これにより、情報構造指向アプローチに基づく既存領域知識モデルに対して高次化に伴う拡張を加える方法に関する知見を事例的に示した。第 V 部では、9 章にて本論文の貢献と限界についてまとめ、10 章にて近年の関連研究を改めて踏まえながら、結論と課題について述べた。

このように本論文では、情報構造指向アプローチに基づく ITS の汎化を目的として多角的な発展を試みた。これらの成果によって、情報構造指向アプローチに基づく ITS が、プログラミング等のより未成熟かつ複雑な領域においても構築でき、構築した領域知識モデルが一定の情報を持っていれば他の教授活動に再利用可能であり、さらに状況に応じて領域知識モデルの複雑化だけでなく、高次化に伴う拡張・変更が可能であることを事例的

に示すと同時に，これらを実現するための方法論に関する知見を示唆したといえる．

キーワード

情報構造指向アプローチ，知的学習支援システム，適応的フィードバック，漸進的学習，
Artificial Intelligence in Education

Multidirectional Development for Generalization of Intelligent Tutoring System Based on Information-structure-oriented Approach*

Kento Koike

Abstract

In Intelligent Tutoring Systems (ITSs), it is essential to describe the models that form the premise of learning activities for realizing intelligent behavior. In general, the main components of ITSs are domain models, learner models, and pedagogical models. In particular, the quality of the domain model determines the extent to which an ITS provides adaptive support. One of the ways to design domain models is the information-structure-oriented approach. This approach is a model-building method that describes the domain knowledge used in the learning task in a computer-readable way. With the domain model based on this approach, the system can perform diagnosis, inference, and feedback in the context of the learning task. In particular, this thesis focuses on this approach to designing a model in which domain knowledge is segmented into components and specified as a set of such components.

In this thesis, we have developed three research directions to generalize ITS based on the information-structure-oriented approach: First, by building a domain model for programming learning, we provided insights into building domain models for relatively immature and complex learning domains. Second, by building a domain-independent pedagogical model for learning-by-teaching, we realized a reusable and domain-independent pedagogical model for existing domain models. We provided insights on how to reuse the model and which models can be reused. Third, by extending existing domain models and introducing metacognition-driven learning, we provided insights into directions for developing existing domain models according to changes in learning domains.

*2022 Dissertation, Department of Electronics and Information Technology, Graduate School of Engineering, Tokyo Polytechnic University, February 1, 2023.

From the above challenges, in domain models based on this approach, results indicated three insights: First, it can be constructed even in immature and complex domains such as programming. Second, it can be reused in other pedagogies if the constructed domain models have certain information. Finally, it can be extended and modified according to requirements to increase the complexity and the level of abstraction.

Keywords:

Information-structure-oriented approach, intelligent tutoring systems, adaptive feedback, progressive learning, artificial intelligence in education.

目次

図目次	xi
表目次	xiii
本論文に含まれる主要な成果	1
第 I 部 導入	3
第 1 章 序論	4
第 2 章 研究の目的とアプローチ	10
2.1 研究の目的	10
2.2 各課題へのアプローチと本論文の構成	12
2.2.1 課題 1: 未知の対象領域に対する領域知識モデル構築法の実現	13
2.2.2 課題 2: 領域知識モデルの汎用的な再利用法の実現	17
2.2.3 課題 3: 領域知識モデルの高次化を伴う汎用的な拡張法の実現	19
第 II 部 未知の対象領域に対する領域知識モデル構築法の実現	22
第 3 章 部品の段階的拡張手法と学習支援システム	23
3.1 本論文における本章の位置付け	23
3.2 プログラミング学習支援としての本章の動機	23
3.3 先行研究	25
3.4 提案手法	26
3.5 提案システム	27
3.5.1 ブロック	27
3.5.2 インタフェースと学習方法	29
3.5.3 段階的な発展	29
3.5.4 フィードバック	30
3.6 評価実験 I	31
3.6.1 実験概要	31

	3.6.2	テスト結果と考察	33
3.7		評価実験 II	35
	3.7.1	実験概要	35
	3.7.2	テスト結果と考察	37
	3.7.3	評価実験 I との統合的な分析	39
	3.7.4	アンケート・インタビュー結果と考察	40
3.8		関連研究と議論	41
3.9		本章のまとめ	43
第 4 章		機能・振舞い・構造の観点に基づく問題解決過程のモデル	44
4.1		本論文における本章の位置付け	44
4.2		プログラミング学習支援としての本章の動機	44
4.3		プログラミングにおける知識の再利用性	47
4.4		部品知識の獲得・拡張活動のプロセス	50
	4.4.1	部品の定義	50
	4.4.2	部品の構築に関する考察	53
	4.4.3	部品知識の獲得・拡張活動プロセスモデル	55
4.5		提案モデルの利用に関する考察	60
	4.5.1	部品の管理機能	60
	4.5.2	構造（ソースコード）の振舞い捕捉機能	61
	4.5.3	誤り箇所の同定機能	62
	4.5.4	振舞いに基づく機能の同定機能	63
4.6		説明機能に関する評価実験	64
	4.6.1	実験方法	65
	4.6.2	実験結果と考察	65
4.7		議論	70
	4.7.1	本章の位置付け	71
	4.7.2	今後の展望	72
4.8		本章のまとめ	74
第 5 章		習熟度を考慮した適応的フィードバック機能	77
5.1		本論文における本章の位置付け	77
5.2		プログラミング学習支援としての本章の動機	77

5.3	機能的階層構造の理解を指向した学習支援	78
5.4	部品の習熟度を用いた適応的フィードバック	80
5.5	初期評価	83
5.5.1	手続き	83
5.5.2	結果および考察	84
5.6	関連研究と本章の位置付け	86
5.7	本章のまとめ	87
 第 III 部 領域知識モデルの汎用的な再利用法の実現		88
 第 6 章 誤りの可視化を重視した Teachable Agent の挙動制御手法		89
6.1	本論文における本章の位置付け	89
6.2	Teachable Agent を用いた学習支援としての本章の動機	89
6.3	関連研究	91
6.3.1	Teachable Agent	91
6.3.2	誤りの可視化	92
6.3.3	先行研究における課題と本章の位置付け	93
6.4	提案手法	94
6.4.1	概要	94
6.4.2	制約	95
6.4.3	初回の挙動制御	96
6.4.4	インタラクションにおける挙動制御	98
6.4.5	対象範囲	101
6.5	ケーススタディ	102
6.5.1	概要	102
6.5.2	乗除算の作問支援における制約	102
6.5.3	制約違反の診断	103
6.5.4	実現可能性の評価	103
6.5.5	TAME に基づく TA の実装	107
6.5.6	有効性の評価	109
6.5.7	全体的な考察	113
6.6	本章のまとめ	114

第7章	ロボットの独自表現である目の発色を用いた学習者の感情制御	116
7.1	本論文における本章の位置付け	116
7.2	ロボットを用いた学習支援としての本章の動機	116
7.3	関連研究	117
7.3.1	教育におけるロボット	117
7.3.2	Intelligent Tutoring System	118
7.4	検証 (A) : 学習者の Academic Emotions を操作できるか?	118
7.4.1	概要	118
7.4.2	目的	118
7.4.3	刺激	119
7.4.4	質問紙	120
7.4.5	手続き	120
7.4.6	結果	120
7.4.7	評価	122
7.5	検証 (B) : 学習者に適応的に Academic Emotions を生み出せるか?	123
7.5.1	個別性を調査するための質問紙	123
7.5.2	方法	125
7.5.3	結果	125
7.5.4	評価	126
7.6	検証 (C) : 実際の講義と同様の状況で, Academic Emotions を操作で きるか?	129
7.6.1	概要	129
7.6.2	目的	129
7.6.3	刺激	129
7.6.4	質問紙	130
7.6.5	手続き	130
7.6.6	結果	131
7.6.7	評価	132
7.7	議論	133
7.8	本章のまとめ	134

第Ⅳ部	領域知識モデルの高次化を伴う汎用的な拡張法の実現	139
第8章	Error-based Simulation と概念マップを用いたメタ認知駆動学習を促すシステム	140
8.1	本論文における本章の位置付け	140
8.2	メタ認知駆動学習支援としての本章の動機	141
8.3	メタ認知とメタ認知駆動学習	141
8.3.1	関連研究	141
8.3.2	本章の位置付け	142
8.4	提案システム	142
8.5	評価実験	144
8.5.1	手続き	144
8.5.2	振り返り記述紙の結果	145
8.5.3	アンケート結果	145
8.6	本章のまとめ	146
第Ⅴ部	総括	147
第9章	本論文の貢献と限界	148
第10章	結論	150
	謝辞	153
	参考文献	155
	受賞一覧	172
	業績一覧	173

目次

1.1	ITS の主要な構成	6
2.1	第 II 部における取り組みと貢献	16
2.2	第 III 部における取り組み	19
2.3	第 III 部における貢献	20
2.4	第 IV 部における取り組みと貢献	21
3.1	金森らによるプログラミングプロセス	25
3.2	提案する「部品の段階的拡張手法」	26
3.3	本システムのインタフェース	28
3.4	スワップの習得画面 (一部)	30
3.5	2 変数ソートの習得画面 (一部)	30
3.6	手順点・構造化点の採点基準の例	33
3.7	評価実験 I: 手順点における基礎・転移課題の評価	34
3.8	評価実験 I: 構造化点における基礎・転移課題の評価	34
3.9	評価実験 II: 手順点における基礎・転移課題の評価	38
3.10	評価実験 II: 構造化点における基礎・転移課題の評価	38
3.11	統合的な分析: 構造化における基礎・転移課題の 3 群の比較評価	40
4.1	プログラミングにおける問題解決過程	46
4.2	機能が再利用可能な例	48
4.3	部品知識の獲得・拡張活動における 理想状態と一般的な学習者	52
4.4	部品知識に含まれる要素間の関係性	52
4.5	Decomposition と Composition の流れ	54
4.6	部品知識の獲得・拡張活動プロセスモデル (「2 変数の入れ替え」を拡張して「2 変数の並べ替え」を獲得する場合を 事例に)	56
4.7	学習者の知識状態の表現	57

4.8	部品知識の獲得・拡張活動プロセスモデル （「2変数の並べ替え」を拡張して「配列の最小値を求める」を獲得する場 合を事例に）	58
5.1	古池らによる学習支援システム	79
5.2	学習者モデルを用いた診断アルゴリズム	82
6.1	提案手法の全体像	94
6.2	システムインタフェース（問題演習フェーズ）	108
6.3	システムインタフェース（教えるフェーズ）	109
7.1	Pepperの目のそれぞれ異なる発色（左から、白、赤、緑）	119
7.2	検証（A）の手続き	121
7.3	発話パターン「重要」における結果	122
7.4	発話パターン「注意」における結果	123
7.5	発話パターン「困惑」における結果	124
7.6	発話パターン「理解」における結果	125
7.7	実験設定	131
7.8	刺激として用いた映像（表7.3における発話10の状況）	132
7.9	検証（C）の実験設定	133
7.10	実験群の結果（エラーバーは標準誤差を示す）	133
7.11	統制群の結果（エラーバーは標準誤差を示す）	134
8.1	提案システムにおけるEBS機能画面	143
8.2	提案システムにおける概念マップ機能画面	144

表目次

3.1	評価実験 I: 基礎課題と転移課題のリスト	32
3.2	評価実験 II: 基礎課題のリスト	36
3.3	評価実験 II: 転移課題のリスト	37
3.4	アンケート項目とポジティブな回答数の比較	41
4.1	比較する説明の例 (説明 3)	66
4.2	評価実験で用いた質問項目の一覧	66
4.3	実験結果の概要 (1: 従来の～5: 提案モデルに基づく説明)	67
4.4	各事例ごとの評価 (1: 従来の～5: 提案モデルに基づく説明)	67
4.5	記述量の上位群と下位群における評価	68
4.6	従来のシステムで提示される説明・B 群におけるクラスタ分析	75
4.7	理由記述文章全体におけるクラスタ分析	76
5.1	学習範囲と事前・事後テストの内容	84
5.2	テストにおける手順点と構造化点の結果 (括弧外は平均, 括弧内は標準 偏差)	85
5.3	システム学習の条件ごとに学習者が受け取ったフィードバック数 (括弧 外は平均, 括弧内は標準偏差)	86
6.1	用意したカードのテンプレート一覧	105
6.2	用意したデータセットの一覧	105
6.3	用意した制約表現の一覧	106
6.4	各テストの結果	110
7.1	発話パターン「重要」における調整スタイルと Academic Emotions の相関	126
7.2	発話パターン「注意」における調整スタイルと Academic Emotions の相関	126
7.3	発話パターン「困惑」における調整スタイルと Academic Emotions の相関	127
7.4	発話パターン「理解」における調整スタイルと Academic Emotions の相関	127
7.5	評価式に基づく調整スタイルと発話パターンの相性	128
7.6	発話スクリプト	136
7.7	発話スクリプト (続き)	137

7.8	発話スクリプト（続き）	138
8.1	アンケートにおける質問項目（6件法）	146

本論文に含まれる主要な成果

本論文に含まれる主要な成果と各部および各章の対応を以下に示す。

- 第II部 3章 [古池 18a]: (ショートノート) 古池謙人, 東本崇仁: プログラミングにおける構造的
理解のための部品の段階的拡張手法の提案とそのシステムの開発, 教育システム
情報学会誌, Vol. 35, No. 2, pp. 215–220, (2018)
- 第II部 3章 [古池 19a]: (原著論文) 古池謙人, 東本崇仁, 堀口知也, 平嶋宗: プログラミングの
構造的理解を指向した部品の段階的拡張手法の提案と支援システムの開発・評価,
教育システム情報学会誌, Vol. 36, No. 3, pp. 190–202, (2019)
- 第II部 4章 [古池 20]: (原著論文 (コンセプト論文)) 古池謙人, 東本崇仁, 堀口知也, 平嶋宗:
プログラミング学習における再利用性を指向した知識組織化のための知的支援: 機
能・振舞い・構造の観点に基づく問題解決過程のモデル, 人工知能学会論文誌, Vol.
35, No. 5, pp. C-J82_1–17, (2020)
- 第II部 5章 [古池 23b]: (ショートノート) 古池謙人, 藤島優希, 東本崇仁, 堀口知也, 平嶋宗: プ
ログラミングの機能的階層構造の学習環境における習熟度を考慮した適応的フィー
ドバック機能の評価, 教育システム情報学会誌, Vol. 40, No. 3, in press, (2023)
- 第III部 6章 [古池 23a]: (原著論文) 古池謙人, 樋村いづみ, 東本崇仁: TAME: 誤りの可視化
を重視した Teachable Agent の挙動制御手法, 電子情報通信学会論文誌 D, Vol.
J106-D, No. 2, in press, (2023)
- 第III部 7章 [Koike 19b]: (原著論文) Koike, K., Tsuji, Y., Tomoto, T., Katagami, D., Obo,
T., Ogai, Y., Sone, J., and Udagawa, Y.: Academic Emotions Affected by Robot
Eye Color: An Investigation of Manipulability and Individual-Adaptability,
International Journal of Advanced Computer Science and Applications, Vol.
10, No. 4, pp. 410–418, (2019)
- 第IV部 8章 [古池 21]: (ショートノート) 古池謙人, 久世泰成, 東本崇仁: Error-based Simu-
lation と概念マップを用いたメタ認知駆動学習を促すシステムの開発と評価, 電子
情報通信学会論文誌 D, Vol. J104-D, No. 6, pp. 526–528, (2021)

この博士論文の3章, 5章は, 同じ著者によって執筆された上記の該当箇所右側に示す文献に基づいており, 教育システム情報学会 (JSiSE) の許可を得て, [古池 18a], [古池 19a] および [古池 23b] から再利用をしている. なお, [古池 18a], [古池 19a] および [古池 23b] の著作権は JSiSE に帰属しており, 当該範囲の内容を利用する場合は JSiSE の規定する著作権規定を遵守されたい.

この博士論文の4章は, 同じ著者によって執筆された上記の該当箇所右側に示す文献に基づいており, 人工知能学会 (JSAI) の許可を得て, [古池 20] から再利用をしている. なお, [古池 20] の著作権は JSAI に帰属しており, 当該範囲の内容を利用する場合は JSAI の規定する著作権規定を遵守されたい.

この博士論文の6章, 8章は, 同じ著者によって執筆された上記の該当箇所右側に示す文献に基づいており, 電子情報通信学会 (IEICE) の許可を得て, [古池 23a] および [古池 21] から再利用をしている. なお, [古池 23a] および [古池 21] の著作権は IEICE に帰属しており, 当該範囲の内容を利用する場合は IEICE の規定する著作権規定を遵守されたい.

この博士論文の7章は, 同じ著者によって執筆された上記の該当箇所右側に示す文献に基づいており, The Science and Information (SAI) Organization の許可を得て, [Koike 19b] から再利用をしている. なお, [Koike 19b] の著作権は SAI Organization に帰属しており, 当該範囲の内容を利用する場合は SAI Organization の規定する著作権規定を遵守されたい.

第 I 部

導入

第1章 序論

「学習」という言葉から、おそらく多くの人は数学、英語、理科など学校での教科学習のことを連想するに違いない。もちろん、教科学習も重要な学習ではあるが、私たちが学習しなければならないことはそれだけではない。ワープロの操作やコンピュータのプログラミングなどの認知技能、水泳やスケートなどの運動技能、ギターやピアノなど楽器の演奏技能、車の運転技能や料理の調理法など、私たちが日常生活を送るのに必要な知識や技能は、すべて学習によって習得されるのである。また、「学習」という言葉からは、通常望ましいものだけを連想しがちであるが、私たちは喫煙のような健康を害する習慣や、盗癖のような法に触れる行為さえも学習することができる。

森敏昭ら (2011) [森 11, 1 章冒頭]

近年の科学技術の発達に伴い、人々の「学習」を取り巻く環境も変化している。「学習」とは、ヒトの最も基本的かつ強力な技能である。我々ヒトは幼児期から体の動かし方や言語を学ぶように、常に学習を繰り返しながら成長する。その中で、ヒトの学習を促すための環境が「学習環境」であり、学習・教育一般において今日も重要な役割を果たしている。学習環境とは、「何らかの知識や技能の獲得が促進される場」と広く定義できる。我々ヒトの教育水準を担保するにあたっては、教授者と学習者の関係が1対Nとなる学習環境における集団教育が存在感を示してきた。このような学習環境は、日本では「寺子屋」を先駆けとして、現在の小・中・高の学校、大学や専門学校等の高等教育機関へと繋がってきている。一方で、集団教育が主流化するにつれ、個々の学習者に沿った適応的な指導の重要性が改めて主張されるようになってきている。これらの一因には、教授者と学習者の数的な非対称性により、学習者に対して教授者が不足していることが挙げられる。そこで、教授者不足のなかで適応的な個別指導を実現する一つのアプローチが約半世紀の間に発展してきた。それは、知的な計算機による学習環境の実現である。

計算機による知的な学習環境の実現を目指して、1950年代から、計算機に教育の一部を補助させる試みとして Computer Aided/Assisted Instruction (CAI) が研究されてきた。そして1970年には、実際に計算機単独で学習者への対象の教育を完結させる学

習環境の最初の事例として、Carbonell による SCHOLAR[Carbonell 70b] が発表された。SCHOLAR は、Quillian の意味ネットワーク [Quillian 68] に影響を受けて開発された知的な推論機能を持つシステムである。このような学習環境は、「知的 (intelligent) な振舞い」によって学習を支援する計算機システムという意味で今日では知的学習支援システムないし Intelligent Tutoring System (ITS) と呼ばれている。SCHOLAR の誕生から 17 年後、1987 年には、Wenger が、その著書「Artificial intelligence and tutoring systems: computational and cognitive approaches to the communication of knowledge」[Wenger 87] において、当時初めて、ITS に関する研究の全体像を示した。下記に掲げる一節は、その Wenger の著書の日本語版において、監訳者序文として示されたものである。

ITS 研究は優れた教授者の振舞いをコンピュータベースで実現すること、良質の知的 CAI^{*1}プロダクトを AI 技術によって効果的に開発できうる環境を確立すること、そして、新しい教育の方法論の探求でもあるといえよう。すなわち、記憶、思考、理解、認識といった不明瞭な事象を明瞭な形式で（たとえそれらが仮説的構造モデルであったとしても）記述し、それをさまざまな推論のしくみのもとでシミュレートし、教育という現象や振舞いの正当性を積み上げていこうとするものである。

Etienne Wenger (1987) [Wenger 87, 日本語版監訳者序文]

ITS において、学習活動の前提となるモデルを記述することは重要である。ここでは、こういった ITS のなかで、「記憶、思考、理解、認識といった不明瞭な事象を明瞭な形式で（たとえそれらが仮説的構造モデルであったとしても）記述」することの重要性が主張されていると捉えられる^{*2}。一般に ITS の主要な構成要素を担うモデルとしては、3つのモデルが挙げられる [大槻 86, Nwana 90, 大槻 00, Nkambou 10, Woolf 10, Sottolare 16]。それは、領域知識モデル (Domain model)、学習者モデル (Learner model)、教授モ

^{*1}知的な Computer Aided/Assisted Instruction のこと。ITS と同義。

^{*2}同時に、こういった「作ることで理解する」(Analysis-by-Synthesis) アプローチは、人工知能や認知科学における基本的な立場の一つである。また、その考えは、ITS だけでなく、その他にも、人工知能や認知科学にルーツを持つ計算論的神経科学や知識工学などの多くの派生・関連分野にも脈々と受け継がれてきている。

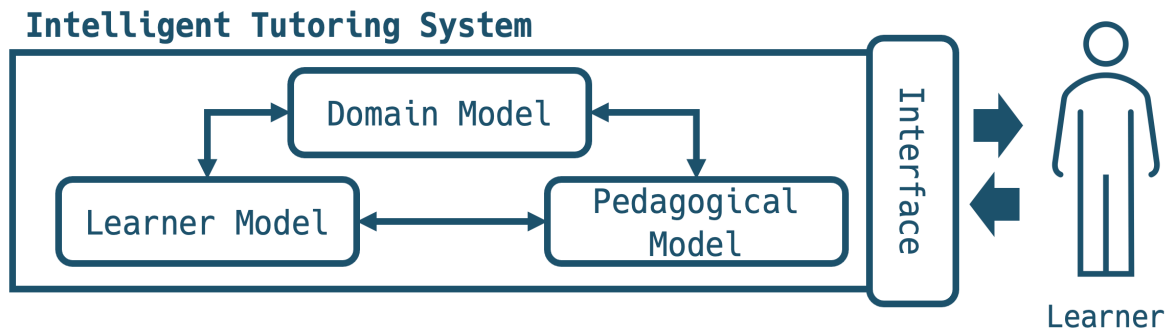


図 1.1 ITS の主要な構成

デル (Pedagogical model) である (図 1.1). 領域知識モデルとは ITS が教授したい対象領域の知識そのものに関するモデルである. 学習者モデルとは, システムが学習者の状態を捉えるためのモデルであり, しばしば領域知識モデルのオーバーレイ (サブセット) として表現される [Sottolare 16] (e.g., knowledge tracing [Corbett 94]; model tracing [Heffernan 08]; bayesian networks [Zapata-Rivera 04]; [Chrysafiadi 13] に詳しい). また, 近年では, 学習者の知識状態だけでなく, 認知状態 [Yamamoto 21] や感情状態 [D’Mello 07, Matsui 19] も表現し, 支援する試みがある. 最後に, 教授モデルとは, その名の通り ITS が学習者にどのように知識を伝達するかといった教授方法論についてのモデルである. 領域知識モデルと学習者モデルを組み合わせると, 学習者モデルが示すその学習者の状態に適応して, 領域知識モデルに基づき何を教えるかを決定するのが主たる仕事となる. これら 3つの内部モデルを用意し, それらに対して, 学習者とのインタラクションを可能にするインタフェースを加えることが ITS の主たる設計指針の一つである.

特に, 領域知識モデルは学習者の状態表現や教授内容の根幹を担うことから, ITS にとって必要不可欠な要素である. ITS がどのような領域知識の構造を持っているかによって, ITS の支援可能な範囲が左右される. 領域知識モデルの構築アプローチは, 大きく次の 2つに分かれる. それは, データ駆動 (data-driven) アプローチと, モデル駆動 (model-driven) アプローチである. データ駆動アプローチでは, 大量のデータに対して統計処理や機械学習の技術を用いることで, 低コストを重視したモデルを構築する (e.g., [Sottolare 16, Chapter 12]). モデル駆動アプローチでは, システムに与える「対象の知

識」を熟達者 (expert) の力を借りるか、学習対象に対する工学的分析を用いて記述することで、システムが適応的な挙動を生成することを重視したモデルを構築する。それぞれのアプローチには、いくつかの課題もある。例えば、前者のアプローチに基づくシステムは、データの収集が課題となり、また、モデルの生成が確率的に行われるため学習者に対する適応性において挙動の精彩を欠くことになる。一方、後者のアプローチでは、記述するモデルの負荷が高く、文脈・状況依存的 (ad-hoc) になりがちであるため、再利用性に欠ける。ゆえに、ITS 研究においてモデルの構築アプローチを選択する際には、これらのトレードオフを引き受けることが要求される。

モデル駆動アプローチには、さらに「アドホックフレーム指向アプローチ (ad-hoc frame oriented approach)」と「情報構造指向アプローチ (information-structure-oriented approach)」に分けられる [Carbonell 70a, 平嶋 13, 平嶋 15, 平嶋 18, 堀口 20]。アドホックフレーム指向アプローチとは、領域知識を自然言語を用いて (計算機が理解できない形で) 表現したり、学習課題の診断を網羅的なルールベースで記述する ad-hoc なモデル構築手法のことである。Carbonell は、それまでに行われてきたアドホックフレーム指向アプローチではシステムが学習者の行動 (解答や問い合わせ) に対して柔軟に対応できないことを批判し、情報構造指向アプローチを提唱した [Carbonell 70a]。情報構造指向アプローチとは、システムが学習課題の文脈に沿った診断・推論・フィードバックを実施するために、学習課題に用いられる領域知識を計算機可読な状態で記述するモデル構築手法である。さらに情報構造指向アプローチでは、構築した領域知識モデルに応じて他の 2 つのモデルである学習者モデルと教授モデルを設計する。これにより、情報構造指向アプローチに基づく ITS は、計算機による知識を用いた推論を基盤として、学習者の行動に知的に応答することを可能としている。

また、平嶋らは情報構造指向アプローチに基づく領域知識モデルを活用した学習課題の設計法として、オープン情報構造アプローチを提唱している [平嶋 18, 平嶋 19]。オープン情報構造アプローチとは、学習者にシステムが持つ情報構造を操作可能対象として開示することで、学習者の活動とシステムが持つ知識を 1 対 1 で対応させることで全ての学習者の行動の意味をシステムが理解して対応することを実現するアプローチである。このような、システムが持つ内部表現とユーザが参照する外部表現を一致させることの重要性に

については、Wenger も指摘している [Wenger 87]. 領域知識モデルにおいては、対象の課題における解のバリエーションを表現する状態空間（問題解決空間）が肝要となる．オープン情報構造アプローチでは、領域知識が構成要素に分節化され、その集合として規定されていることが前提となっている．これは同時に、学習者の解を構成要素の有限個の組み合わせとして表現できることを意味しており、解の状態空間を閉空間として表現することを可能にしている．よって、学習者モデルで学習者の解答が、最終的な解（正解）に対してどのように位置付くのかを進捗度合いとして表現したり、それを用いることで教授モデルにおいて学習者の解に応じたフィードバックを（ad-hoc に記述することなく）網羅的に設計することが可能になっている．本論文では、情報構造指向アプローチのうち、オープン情報構造アプローチに耐えうる領域知識モデル（領域知識を構成要素に分節化し、その集合として規定するモデル）を設計する方法を特に「強い情報構造指向」と呼び、単に何らかの領域知識モデルを作ればよいという「弱い情報構造指向」と区別して定義する*3．以降、本論文における情報構造指向アプローチは、特別な記述がない場合この「強い情報構造指向」を指すこととする．

情報構造指向アプローチに基づく代表的な ITS としては、平嶋らの研究グループにおける ITS 「モンサクン」に関する一連の研究（cf., [平嶋 19]）がある．モンサクンの場合、実現されている一連の学習活動は、全て最初期に提案された領域知識モデルである MIPS [平嶋 92] に原点をみることが出来る．MIPS は、つるかめ算を事例として算数文章題に関わる問題解決過程を計算機可読な表現として定式化した領域知識モデルであり、このモデルを活用した学習支援 [中野 00, 中野 02, 横山 06, 横山 07, 倉山 12, 山本 14, 前田 15, 山元 17, 山本 18] から、モデルの再定式化および拡張 [Hirashima 14, 平嶋 15],

*3 ITS 開発に資する学問の一つであるオントロジー工学では、「存在そのものを解き明かす」ことを目指した知識記述法として Ontology を挙げているが、そのうち「存在の厳密な記述」を目指したものを「Heavy-weight Ontology」（オントロジー工学の対象）、一方で「メタデータや単なる語彙の集合」であり概念体系を示さないものを「Light-weight Ontology」と呼んでいる [溝口 05]．本論文の「強い情報構造指向」と「弱い情報構造指向」の区別はオントロジー工学におけるこの2つの差異を近似できると考えている．つまり、「強い情報構造指向」は、「学習活動に用いられる領域知識」の厳密な定義を指向したものと見え、「弱い情報構造指向」は、厳密さや妥当性は問わず、単に推論できる媒体の追加を指向したものと考えられる．一方で、これら2つの間は離散的ではなく、連続的なものであると考えられるため、指向性以上の厳格な区別は困難であろう．また、オントロジー工学においても「真理は中庸である」[溝口 05]と述べられているように、厳密さのみを追究しては現実的な開発や利用には耐えられず、一方で、簡便な記述にしてしまえば「学習」の支援は困難となるため、堀口らも指摘 [堀口 20] するように、「強い情報構造指向」のうえでも、適切な粒度や抽象度で表現することが肝要となる．

モデルに基づく学習活動の分析 [神戸 15, Supianto 16, Supianto 17b, Supianto 17a, Hasanah 17, Hasanah 19] など, 今日までのモンサクンに脈々と受け継がれている. また, 近年ではモデルと学習活動の分析に基づいて簡易脳波計 (EEG) を活用することで「Monsakun Affective」といった, 学習者の行き詰まりを検知して支援を与える ITS が実現されている [Yamamoto 21]. さらに, モンサクンに関する一連の研究で追究された作問学習の情報構造化は, 計算機の外側における作問学習の議論にも影響を与えている [竹中 18].

このように, 情報構造指向アプローチに基づく ITS は, 算数文章題だけでなく, 物理の作図描画問題^{*4} などをも含む Open-ended な課題を, 情報構造指向アプローチに基づく情報構造化 (モデル化) によって Closed-ended な課題に変更 [林 17] することで, 有意義な演習を実現している.

^{*4}物理の範囲においても, 情報構造指向アプローチに基づく ITS 開発が盛んである (e.g., [今井 08, 篠原 16, 山田 16, Ueno 19, 植野 20, Hirashima 16, 堀口 20, 堀口 08, Horiguchi 14, Hirashima 17, 篠原 17, 相川 20]).

第2章 研究の目的とアプローチ

2.1 研究の目的

学習課題に対するこのような活動の設計を可能する情報構造は、その学習課題の特性を的確に反映したものでなければならない。このため、つくられる情報構造は、学習課題に依存したものとなる。しかしながら、その情報構造に基づいてさまざまな学習活動やシステムの機能を設計できることから、学習活動やシステム機能の設計に対しては一般性をもつといえる。例えば、ある学習課題を適切に情報構造化できれば、その情報構造に基づいて、解く、組み立てる、変える、比べるなどの学習活動を、一貫性をもって設計することが可能になる。また、システムによる情報構造の診断機能に関しても、正解となる情報構造を設定すれば、正誤や正解とのずれを、情報構造の一致・不一致および不一致における過剰と欠落としてシステムティックに判定可能となる。さらにいえば、個々の情報構造は個々の学習課題に依存することになるが、それらに対して設計される学習活動や、システムによる診断などの機能は、ある程度の一般性をもち得る。例えば、ある学習課題に対する、つくる、という学習活動は、他の学習課題においても概ね成立するであろうし、その診断に関しても、基本的な方法は同じとなり得る。つまり、構造自体は特殊性をもつことになるが、構造に対する操作に関しては、一般性をもち得ることが期待できる。

平嶋宗 (2015) [平嶋 15, 1 章]

情報構造指向アプローチに基づいて開発された ITS では、対象領域に特化した領域知識モデルが活用される [Carbonell 70a, 平嶋 13, 平嶋 15, 平嶋 18, 堀口 20]。これにより、ITS は対象領域の知識や特有の誤りに踏み込んだ適応的支援を網羅的に設計することが可能になる。一方、このような ITS においては、領域知識モデルを対象領域に特化して作り込む必要があることから、対象領域の緻密な分析が必須となる。ゆえに、何を学ぶべきかが自明ではなく解も一意に規定しえないような対象領域に対しての開発が困難である。それに加えて、領域知識モデルを独立して再利用や拡張をするための知見が少ないこと

が、ITS1 つあたりの開発コストを高めている。平嶋は、情報構造指向アプローチに基づく ITS の汎用性について、冒頭のように述べている [平嶋 15]。特に、「個々の情報構造は個々の学習課題に依存することになるが、それらに対して設計される学習活動や、システムによる診断などの機能は、ある程度の一般性をもち得る」と述べているように、情報構造指向アプローチに基づいて作られた領域知識モデルの独立性やモジュール性など、その汎用性を示唆する指摘がなされている。ここで汎用性とは、ある ITS のために作られた領域知識モデルを、その ITS に含まれる学習者モデルや教授モデルから切り離し、独立して他の学習者モデルや教授モデルに適用できることを意味する。つまり、汎用性を有するモデルは、各構成モデル間の組み合わせを変更することで、別の学習文脈や学習効果を発揮することができ、モデルの効率的な再利用ができることを意味する。このことは、情報構造指向アプローチに基づく ITS の開発コストが高いという欠点を補う。しかしながら、実際に当該のモデルを再利用ないし他の教授モデルに適用するなどといった情報構造指向アプローチの汎用性を示した研究は見当たらない。よって、情報構造指向アプローチに基づく ITS の開発が汎化されているとはいわずらいのが現状である。

そこで、本論文では情報構造指向アプローチに基づく ITS の汎化を目的として、以下に述べる 3 つの課題を設定し、多角的な発展を試みた。1 つ目の課題は、従来研究と比較して未知の対象領域における領域知識モデルの構築法に関する知見を創出することである。これまでは、領域知識モデルの構築が比較的容易な数学 (e.g., [平嶋 19]) やニュートン力学 (古典力学) (e.g., [Hirashima 16, 堀口 20]) など、ある程度既知な領域を中心に情報構造指向アプローチに基づく ITS が開発されてきた。一方で、より未知な領域、すなわち未成熟かつ複雑な対象領域に対する情報構造指向アプローチに基づく ITS は、未だに事例が多くない。よって、このような ITS の実現は情報構造指向アプローチの対象とする領域の拡張に繋がり、情報構造指向アプローチに基づく ITS の汎化に寄与する。

2 つ目の課題は、既存の領域知識モデルを再利用可能かつドメイン独立な教授モデルを実現し、モデルの再利用法および再利用可能なモデルについての知見を得ることである。先述のように、情報構造指向アプローチに基づく領域知識モデルは、再利用に関する知見がなく、ほとんど再利用されていないという課題が存在する。そこで、情報構造指向アプローチに基づく既存の領域知識モデルが再利用可能であることや、再利用性の高い領域知

識モデルの特徴と再利用するための領域独立な教授モデルに関する知見を創出することは、長期的な開発コストの低減に繋がり、情報構造指向アプローチに基づく ITS の汎化に寄与する。

3つ目の課題は、既存の領域知識モデルにおける学習対象の高次化に伴う拡張における知見の創出である。情報構造指向アプローチに基づく ITS の汎化を議論するうえで、情報構造指向アプローチに基づく領域知識モデルそのものが拡張可能であるかどうかは、システム設計者にとって重要である。なぜなら、教授意図に沿って領域知識モデルが対象とする範囲を変化させたいという要求は容易に想像でき (e.g., 算数 ITS において「加減算の範囲から乗除算へ拡張したい」), そのような要求ごとに領域知識モデルを 1 から再構築することになれば、開発コストが増大してしまうためである。そこで、学習対象自体が変化したときにおける拡張可能性の向上を目指すことは、多様な教授意図に基づいた既存の領域知識モデルを変更を促し、情報構造指向アプローチに基づく ITS の汎化に寄与する。

このように本論文では、情報構造指向アプローチに基づく ITS の汎化を目的として多角的な発展を試みる。これらの成果によって、情報構造指向アプローチに基づく ITS が、プログラミング等のより未成熟かつ複雑な領域においても構築でき、構築した領域知識モデルが一定の情報を持っていれば他の教授活動に再利用可能であり、さらに状況に応じて領域知識モデルの複雑化だけでなく、高次化に伴う拡張・変更が可能であることを事例的に示すと同時に、これらを実現するための方法論に関する知見を示す。

2.2 各課題へのアプローチと本論文の構成

本論文では、第 I 部で導入を行い、序論と本論文の目的について述べてきた。ここからは、第 II 部、第 III 部、第 IV 部にわたって、各部でそれぞれの課題にアプローチする。そして、最後に第 V 部にて総括のため結論を述べる。

第 II 部 未知の対象領域に対する領域知識モデル構築法の実現: 未成熟かつ複雑な対象領域における領域知識モデル構築法に関する知見創出

第 III 部 領域知識モデルの汎用的な再利用法の実現: 既存の領域知識モデルを再利用可能かつドメイン独立な教授モデルの実現と、モデルの再利用法および再利用可能なモデ

ルについての知見創出

第IV部 領域知識モデルの高次化を伴う汎用的な拡張法の実現: 既存の領域知識モデルにおける学習対象の高次化を伴う拡張法に対する知見創出

2.2.1 課題 1: 未知の対象領域に対する領域知識モデル構築法の実現

情報構造指向アプローチに基づく ITS の一つである「モンサクン」(cf., [平嶋 19]) が対象とする学習領域である数学は、学習対象領域のなかで最も既知であり成熟している分野の一つであるといえる。とりわけモンサクンの範囲である小領域としての算数文章題については、文章表現と量的表現とのマッピングが必要となり、必ずしも十分に整理されている領域とはいえないが、上位となる数学が成熟していることによる恩恵は大きいものと推察される。一方で、より未知な領域、すなわち未成熟で複雑な対象領域においては、情報構造指向アプローチに基づく学習環境としてモンサクンのように一連のプロジェクトとして成功している事例は見当たらない。この要因の一つに、そのような未成熟な領域では MIPS[平嶋 92] のような基盤となるモデルの構築が困難であることが考えられる。これは、対象領域自体が未知であるがゆえに、解法が一意に規定できず、また、解の正しさ機械的に診断できないためである。そこで第II部では、このような、相対的に未成熟かつ複雑であるがゆえに未知な対象領域を対象とした領域知識モデル構築法についての知見創出を目的とする。具体的には、相対的により未成熟かつ複雑な領域 [Hicks 14, Rivers 17] としてプログラミングを取り上げ、プログラミング学習を対象に情報構造指向アプローチに基づく ITS の実現を目指した一連の研究を行った。21世紀スキルや Computational Thinking[Wing 06]などでプログラミングが取り上げられるように、プログラミングの学習と問題解決能力の学習に一定の関連性が認められている。この要因の一つは、プログラミング自体が「紙とペン」と同じように問題解決に用いる道具立てとしての特徴を有しており、また、問題解決の「手順」に深い注意を払う必要があるためと考えられる。そこで、未知領域のうち特にプログラミング学習を対象とすることで、あらゆる学習対象に含まれる「問題解決」自体の構造化に寄与する汎用的な知見が得られると考えた。

プログラミング学習は、特に以下に懸念される大きく2つの困難性から、上記のような

学習環境に適用できるモデルを構築するに至っていないのではないかと考えている。

1. プログラミング学習で何を学ぶべきかについての議論が複数存在し [Milne 02, Luxton-Reilly 18], 中には悪定義 (ill-defined) 領域 [Lynch 09] をも含む [Le 13].
2. プログラミングとは, 現実世界に存在する「何か」を操作するいわば道具であり, 「○○を対象としたプログラミング」という形で領域が入れ子構造になるため, 概念化が困難である.

Lynch は, 悪定義問題と悪定義領域について, 次のような定義をしている [Lynch 09].

- 問題が悪定義 (ill-defined) であるとは, 本質的な概念, 関係, 解決基準が未確定または過小であり, 文脈が不明確または難解で, 解決者が問題の枠組みを作るか再特定することを必要とする場合である.
- 悪定義領域 (Ill-defined domains) とは, その領域の問題解決策や事例を検証する手段を提供するための, 本質的な概念, 関係, 手順を一意 (unique) に特定する単一の強力な領域理論を欠いている領域である. そのため, 問題解決者は領域内で問題解決する際に, 領域を構造化するか再特性化する必要がある.

プログラミングはこの意味で, 悪定義領域であるといえる. また, Le ら [Le 13] は, プログラミングにおける良定義問題と悪定義問題の例として, 以下のような事例を挙げている.

- 2つの数字, 4 と 5 を足し合わせる Java 構文を記述するために, 空欄部の演算子を埋めなさい:
$$S = 4 \text{ ______ } 5;$$
- 投資シミュレーションシステムを開発しなさい

前者は良定義問題であり, 解が一意に決定する (ちなみに答えは「+」であり, 演算子の選択肢は限られている). 後者は悪定義問題であり, どのように解くべきか (開発すべきか) はほとんど無限の方法があると言ってよい.

よって, プログラミングのような悪定義領域 [Lynch 09, Le 13] においては, 適切な粒

度 [堀口 20] を決定できるかどうか領域知識モデルの構築の成否を分かつほど肝要となる。実際、Hicks らは、「プログラミングの ITS は状態-行動のペアを覚えるだけでは表現すべき状態空間が広すぎるためにうまくいかない」ことを指摘している [Hicks 14]^{*5}。したがって、プログラミングの領域知識モデル^{*6}を構築するためには、領域知識モデルの構築コストを前提とした学習活動の選定が不可欠である。そこで1つ目の困難性に対して3章では、プログラミング学習において学ぶべきことに一つとして「知識の再利用性向上」を取り挙げ、それを指向した知識体系化支援に取り組んできた。3章ではこの知識体系化を通して、有意味な一連のコードを部品として獲得し、それを再利用して他の部品と適応的に組み合わせられるようになることを狙っている。そのために、知識の体系化を行う学習手法を提案し、その支援システムの構築および学習効果の検証を行った [古池 18a, 古池 19a]^{*7}。これにより、任意の粒度で学習活動を実現できる学習手法を提案し、必ずしもすべての領域知識を網羅せずとも、任意の抽象度からプログラミング学習の領域知識モデルを構築可能とした。

また、2つ目の困難性に対して4章では、プログラミングとは何か（情報）を操作して処理する機械システムであると考え、定性物理やオントロジー工学で議論されてきた人工物の概念化 [De Kleer 84, 西田 93, 笹島 96, 来村 02b, 来村 02a] に近似することでその解決を試みた。これにより、3章で提案した学習活動の問題解決中に扱われる知識として「部品」を知識記述的に定義し、その部品の獲得や拡張における問題解決プロセスを考察することで学習者の学びを支援するための領域知識モデルを提案した [古池 20]。4章にて

^{*5}それ故、彼らは複数の学習者の問題解決過程をマルコフ過程として収集しておくことで、学習時の学習者の行動をマルコフ過程に当てはめ、それに合わせたヒントを生成する「Hint Factory」を提案している。ただし、このようなデータ駆動アプローチには、先にあげた課題に加え、十分なデータが集まるまでシステムが有用な振舞いをできない「コールドスタート問題」がつきまとう。

^{*6}プログラミング開発における類似の概念として、ドメイン駆動開発 (Domain-driven Development) があるが、この「ドメイン」は先述した「○○を対象としたプログラミング」のうち○○に相当する「プログラミングを道具立てとして解決したい対象」(e.g., 物流管理, 数学, 力学) である。一方、第II部では、道具立てとしてどのようにプログラミングを理解するかを指向している。このため、第II部で指す領域知識 (ドメイン) モデルとは、プログラミングのアルゴリズムに関する知識や機能に関する知識などを指す。通常、プログラミングを行う上では、双方の「ドメイン」への理解が必要となり、この理解対象の混在がプログラミング学習の困難性を高めている一つの要因である。

^{*7}第II部ではあくまでも「ソースコードの有意味な塊」を「機能」と関連付けて理解することに重点を置いており、オブジェクトやクラスといった概念的知識の理解の重要性 [Xinogalos 15] は理解しつつも第II部のスコープとしては取り扱っていない。

問題解決過程に用いる知識を明らかにしたことで、学習者の知識状態を表現することが可能となった。そこで5章では、これまでのシステム [古池 18a, 古池 19a] に対して、部品の習熟度を考慮した学習者モデルを導入することで試験的に適応的なフィードバックを実現した [古池 23b]。これらの一連の取り組みによって、プログラミング学習を対象として領域知識モデルの構築および対応する教授モデルと学習者モデルの構築が可能であること、すなわちプログラミングにおいて情報構造指向アプローチに基づく ITS を開発示すことを示した。

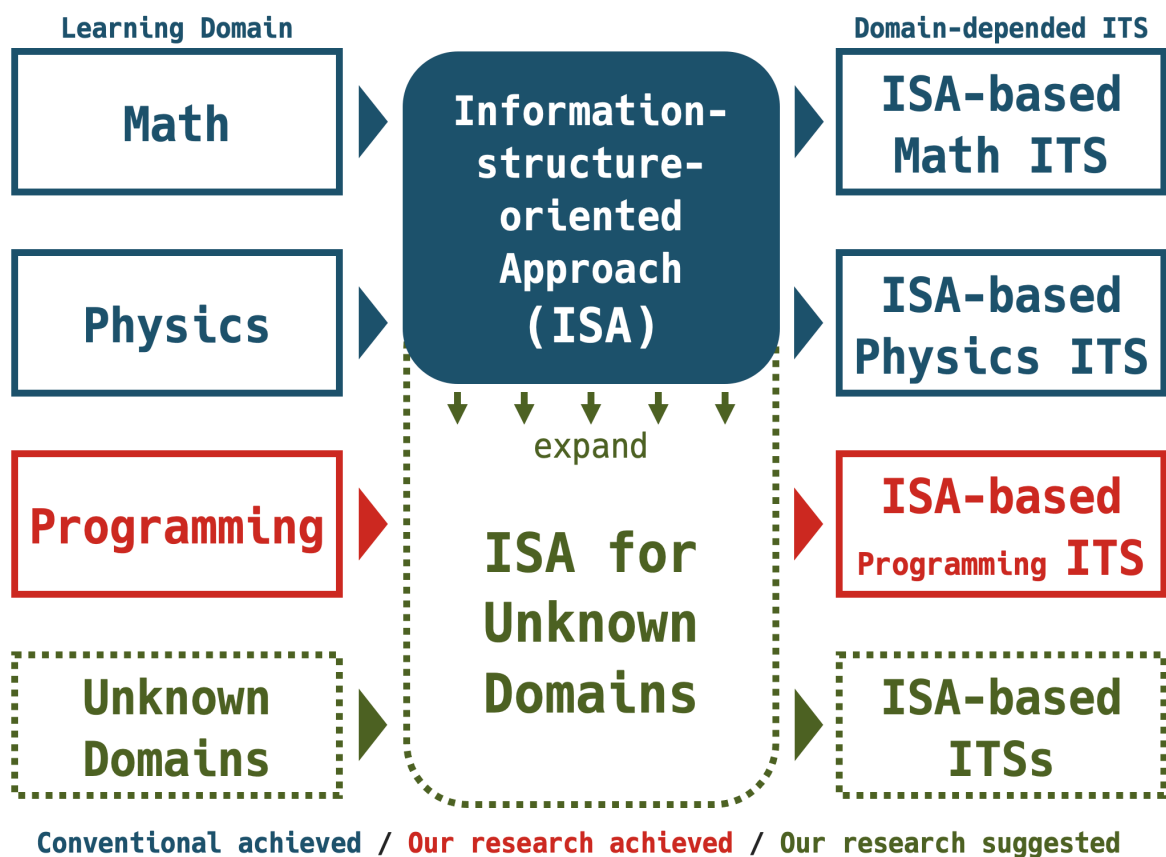


図 2.1 第 II 部における取り組みと貢献

第 II 部の取り組みと貢献の概要を図 2.1 に示す。第 II 部では、これまで数学 (e.g., [平嶋 19]) や、古典力学 (ニュートン力学) (e.g., [Hirashima 16, 堀口 20]) といった既知の領域が対象とされてきたことを取り上げ、従来研究と比較してより未成熟かつ複雑な

対象領域であるプログラミングを対象に情報構造指向アプローチに基づく ITS が開発できることを示した。また、このような成果から、任意の知識粒度を最小単位とする知識組織化を目的とする教授モデルと、知識を機能-振る舞い-構造の単位で表現する領域知識モデルを適用することによる情報構造アプローチの適用法を新たに示した。さらに、この方法を適用できる対象領域においては、未知の対象領域であっても情報構造指向アプローチに基づく ITS を構築しうることを示唆した。

2.2.2 課題 2: 領域知識モデルの汎用的な再利用法の実現

2.1 節で述べたように、これまで情報構造指向アプローチに基づく領域知識モデルの汎用性が主張されていたものの [平嶋 15], (a) 実際に異なる教授モデルに対して再利用できるかどうか, (b) どのような教授モデルであれば領域独立に領域知識モデルと組み合わせられるか, は明らかではなかった。そこで第 III 部では、情報構造指向アプローチに基づく領域知識モデルを用いて他者に教えることによる学び (Learning-by-teaching: LbT) [Duran 17, Fiorella 13] の実現に向けてアプローチする。

ITS に対する LbT を実現するためには、教えられる対象をエージェントとして用意し、そのエージェントが「学習者に教えてもらったことに基づいて問題解決する」という姿を見せる Teachable Agent の採用が有効である [Biswas 05, Leelawong 08, Matsuda 13, 松田 18, Okita 13]。このようなエージェントをメタファーとして用意することで、学習者が ITS を社会的な存在と認知するようになり [Gulz 20, 松田 18, Tärning 19], 教える責任感 [Chase 09] や、学業成績 [Okita 07] などに影響することが知られている。

そこで 6 章では、Teachable Agent の挙動制御を踏まえた教授モデルとして TAME (Teachable Agent Module for Error-visualization) を提案した [古池 23a]。これにより、モンサクンを題材に (a) 情報構造指向アプローチに基づく領域知識モデルが再利用可能であること, (b) 学習者が解答に用いる構成要素とその構成要素間の望ましい関係性 (制約) を用いれば教授モデルを領域独立に設計できること, を示した。同時に、情報構造指向アプローチに基づく領域知識モデルを再利用するためには、解答に用いる構成要素とその構成要素間の望ましい関係性 (制約) の記述が肝要であることを示した。

また、学習者が Teachable Agent を社会的な存在と認知する度合いが LbT の効果を

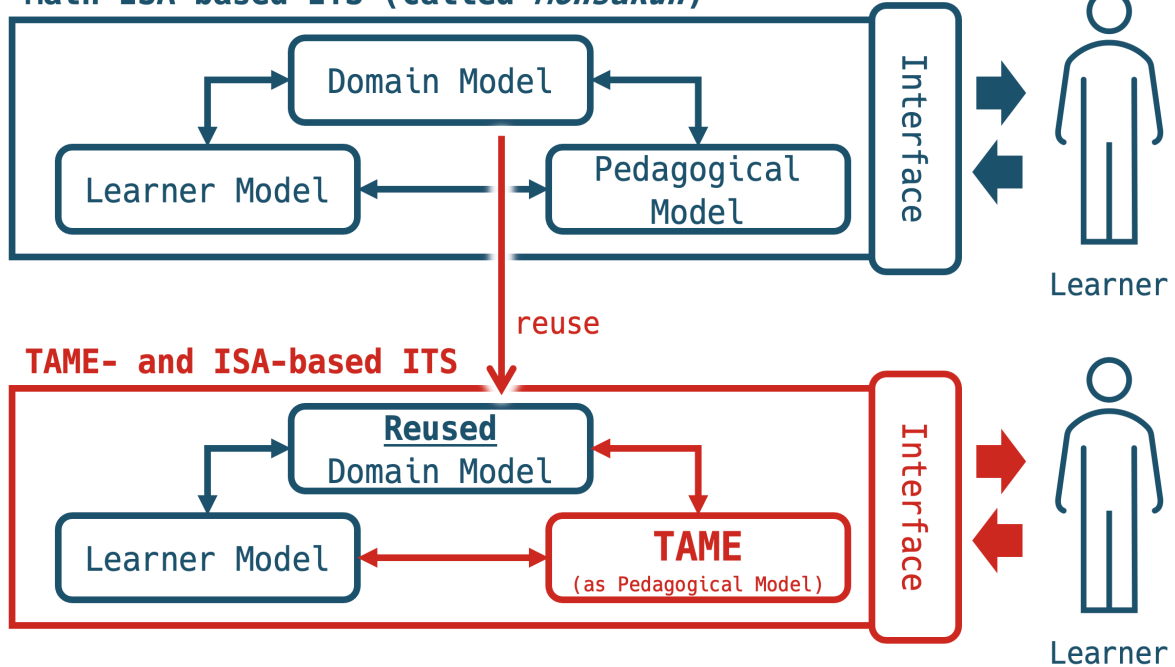
左右すること [Chase 09, Okita 07] を鑑みれば, Teachable Agent の存在感 (presence) を強めたり, コミュニケーション能力を向上させることが重要となる. このための方法の一つとして, バーチャルエージェントからフィジカルエージェント, すなわちロボットへの転換が考えられる. ロボットは, ヒトとして認知される傾向 (擬人化傾向) があることや [神田 08, 柏原 19], 感情や動作の表出によって社会的存在らしさを表現できることが知られており [Leite 14], 学習の文脈で広く活用されてきている [柏原 20, Inazawa 22, Ishino 22, Belpaeme 18]. なかでも, 感情の表出にあたっては, 例えば, ヒトの顔の筋系をロボットで模倣することでヒトと近い感情表出を行う試み [Sakai 17b, 境 17a] がなされている. このようなロボットのヒトに対する類似度を向上させていくアプローチは, まさに擬人化傾向を最大化し, ロボットがヒトとして解釈されることを狙ったものである. しかし, このようなアプローチでは, ヒトが可能な感情表出方法のなかでロボットの感情表出方法を選ぶことになり, 感情表出においてはヒトと同等もしくは下位互換をロボットに期待することになる. よって, ロボット独自の感情表出方法を模索することは, ヒトの mimicking にとどまらない, ロボットならではの学習体験のデザインに寄与するであろう. ロボット独自の感情表出方法の一つとして, ロボットの LED 等を自発光させることによる色を用いた感情表出が挙げられる. 実際, 色がヒトの感情に影響することは広く確認されており (e.g., [Gao 06, Elliot 07]), ここに新たな学習支援の可能性がある. また, 学習支援エージェントが感情を表出するのは, 多かれ少なかれ学習者の状態を affective な方向性から制御することがねらいである.

そこで 7 章では, Teachable Agent がロボット化された際の学習者の感情制御にめがけ, Pepper (Softbank Robotics 社) を題材に, Pepper の目の LED を発色させることによる感情制御手法を模索した [Koike 19b]. これにより, 発話文脈と目の発色の組み合わせによって学習者の感情を制御する手法を提案し, 学習者に対する一定の感情の制御可能性を明らかにした. この手法についても領域独立であり, 学習文脈に依存しないことから, TAME の手法と組み合わせられることはある程度自明であろう.

したがって, 図 2.2 にまとめるように, 第 III 部では, 情報構造指向アプローチに基づく既存の領域知識モデルを再利用可能することで, 新たな教授モデルに適用可能であることを示し, 感情的な支援の可能性を示した. さらに, 図 2.3 に示すように, これらの成果

ISA: Information-structure-oriented Approach

Math ISA-based ITS (called *Monsakun*)



Conventional achieved / Our research achieved / Our research suggested

図 2.2 第 III 部における取り組み

から、領域知識モデルを再利用するための教授モデルに関する汎用的な知見と、どのような領域知識モデルであれば再利用できるかを示唆した。

2.2.3 課題 3: 領域知識モデルの高次化を伴う汎用的な拡張法の実現

既に挙げたように、算数 ITS において「加減算の範囲から乗除算へ拡張したい」というような、既存領域知識モデルの拡張に関する要求は容易に想像できる。実際、「加減算から乗除算へ変える」といった学習対象の複雑化に伴う拡張であれば、平嶋らの「モンサクン」の範囲ですでに実証されている [山元 17, 平嶋 19]。一方、近年では学習における学習者自身のメタ認知 (e.g., 自分がどのように考えて問題解決したかを認知して、考えを変更する) への働きかけが重要視されており [茅島 08, Kayashima 10], そのような教授意

ISA: Information-structure-oriented Approach

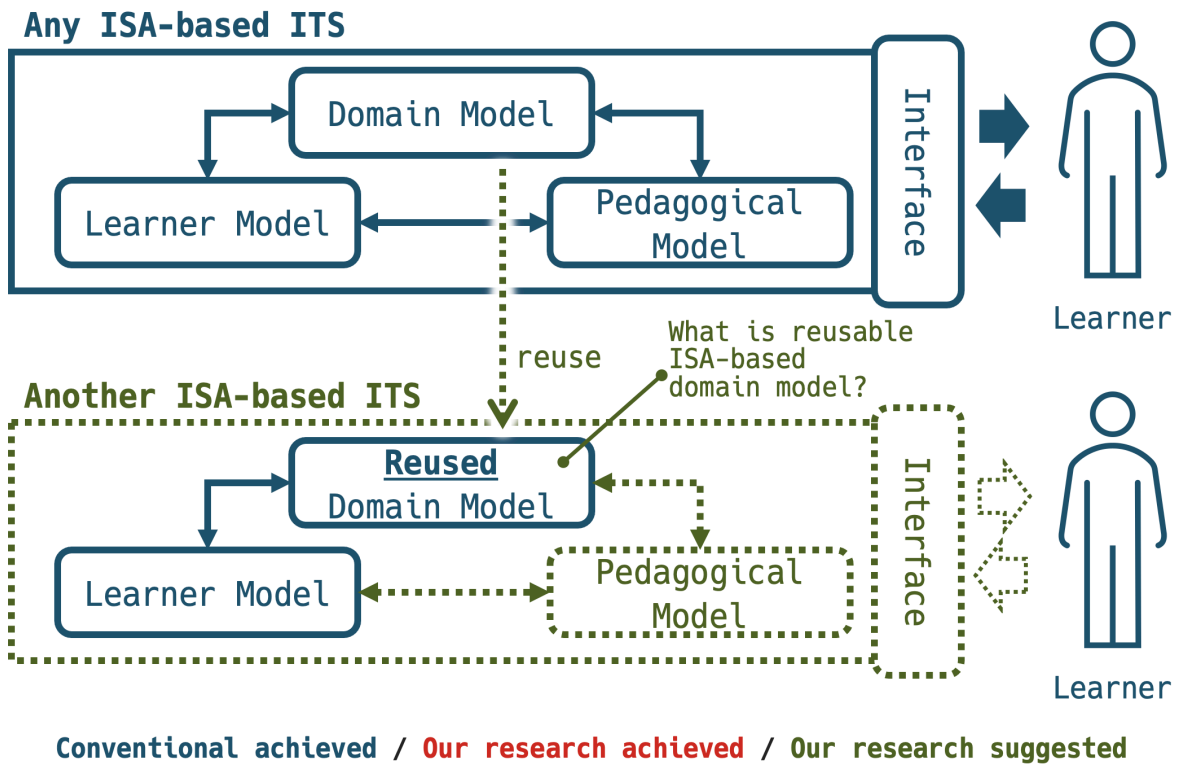


図 2.3 第 III 部における貢献

図への対応も必要となる。しかし現状では、学習対象を「加減算から“加減算における自身の問題解決”へ変える」といった学習対象の高次化に伴う拡張は検証されていない。

そこで第 IV 部では、8 章にて情報構造指向アプローチに基づく既存の領域知識モデルに対し、さらに高次の学習活動であるメタ認知を促すための拡張を行い、その実現可能性を検証した。具体的には、既存の力学 Error-based Simulation システム [Hirashima 98, 平嶋 06, Horiguchi 16, Hirashima 16] の領域知識モデルに対して、茅島らが提唱するメタ認知駆動学習 [茅島 08, Kayashima 10] を実現するための拡張を行った。これにより、図 2.4 に示すように、力学を事例として、情報構造指向アプローチに基づく領域知識モデルの新たな拡張の方向性に対する知見を創出した。

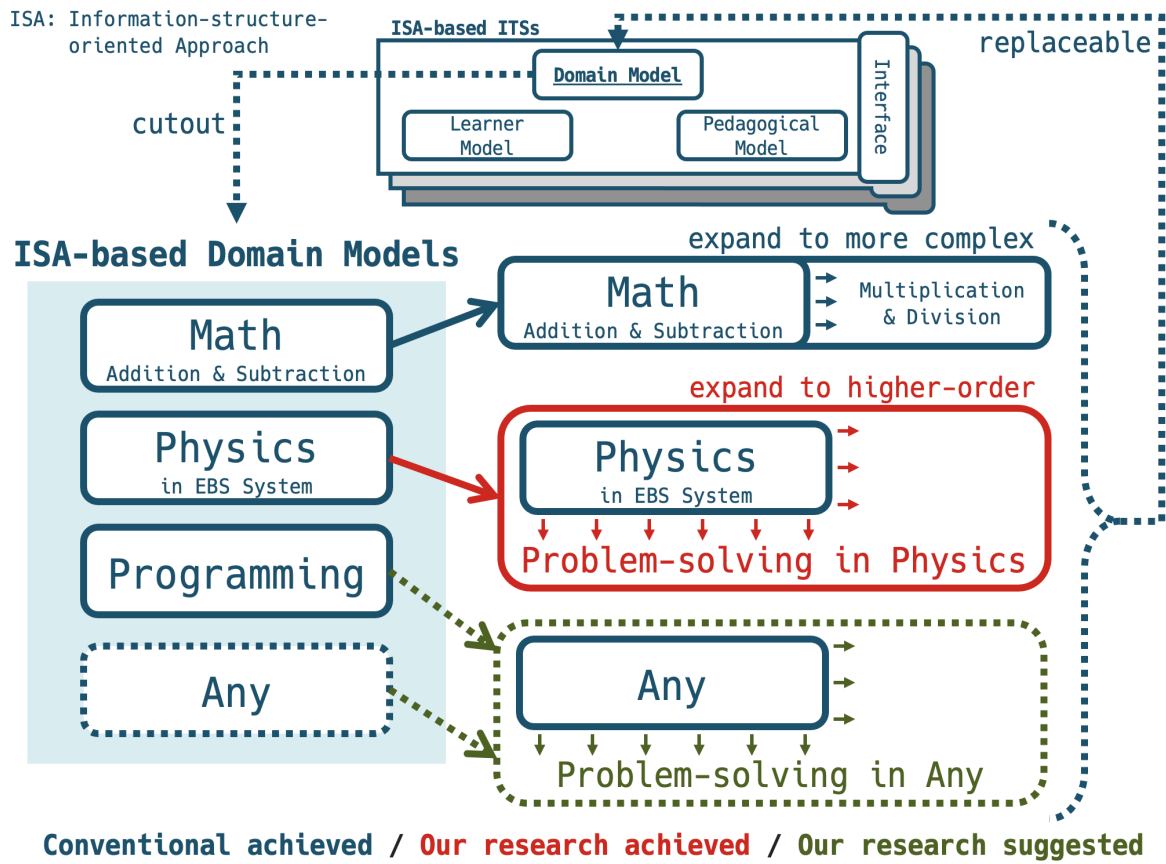


図 2.4 第 IV 部における取り組みと貢献

第II部

未知の対象領域に対する 領域知識モデル構築法の実現

第3章 部品の段階的拡張手法と学習支援システム

3.1 本論文における本章の位置付け

情報構造指向アプローチに基づく ITS の汎化を目指すうえで、第II部では1つ目の課題として、構造が十分に知られていない未知の対象領域においても情報構造指向アプローチに基づく ITS が構築しうることを示すことを目的としている。そのため、第II部では、未成熟かつ複雑であるがゆえに構造が未知の対象領域としてプログラミングを選定し、事例的にプログラミングを対象とした情報指向アプローチに基づく ITS を開発することで、対象領域の汎化を目指す。しかしながら、プログラミングでは、学習対象によって領域知識モデルで表現すべき内容が大きく異なることが想定される。また、2章で述べたように、プログラミングはそもそも何を学ぶべきかが自明ではないという課題がある。そこで本章では、第II部のうち、教授モデルを領域知識モデルに先行して提案することで表現すべき領域知識モデルに示唆を提示する。また、提案する教授モデルは、プログラミングにおける網羅的な領域知識の記述を必要とせず、取り扱う知識を任意の粒度にスケールすることができる。この教授モデルをプログラミングではない領域へも応用することで、同様に網羅的な記述が困難な領域への適用が期待できる。

3.2 プログラミング学習支援としての本章の動機

プログラミングにおいては、if 文や for 文などの様々な処理を組み合わせることで要求された事柄（たとえば、入力データを操作し、期待した出力を行わせるなど）を実現できることが重要である。しかし、「入力 A と入力 B を足して出力する」ということを行う極めて単純なプログラムでも、「足す」処理と「出力する」処理の2つが必要となる。より複雑なプログラムになれば、より多くの処理を要することになる。では、複雑なプログラムを記述する時に熟達したプログラマが処理を一つずつ考えながら記述しているかということについては、そうではない [Shneiderman 79]。例えば、熟達したプログラマがある配列を並べ替えたいと考えた時には、プログラマが今まで作成あるいは参照してきた経験から、自身の知識を検索し、2重 for 文を用いた一連のコードを容易に呼び出すことができる。

これは、熟達したプログラマが配列の並べ替えに必要な一連のコードを並べ替えの機能と関連付けて理解し、それを再利用してきたからであると推測できる。プログラミングにおいては、一連のコードをそのコードが果たす機能として理解し、貯蔵することで他のプログラムに再利用することが可能となる。さらに、大規模なプログラムを理解するときにおいても、機能的に有意義なまとまりとして一連のコードを理解することが重要である。本章では、このような機能として有意義な一連のコードを部品と定義している。

学習者が部品として知識を獲得するためには、一連のコードを明示的に部品として再利用することを繰り返すことで、その一連のコードをまとまりとして把握し、機能と関連付けることが有用であると考えられる。しかし、通常のプログラミング教育においては教師がサンプルコードの提示を行い、学習者が提示されたコードを入力して動かし、その後カスタマイズするという行程を課題形式で行うことが多く、学習者自身が能動的にサンプルコードの意味や仕組みについて考える機会が不足する場合が多く存在することが指摘されている [東本 16]。この場合、提示された課題がサンプルコードを発展させたものであっても、コードのまとまりの意識化は明示的には行われぬ。さらに、学習者が構築した一連のコードを明示的に部品として再利用させる機会は少ない。つまり、学習者にとって通常のプログラミング教育のみでは、部品を獲得する機会が不足していると考えられる。

Shneiderman は、プログラミングの理解において、単一の要素を複合して1つの大きな要素として理解することを繰り返して、構造的な理解に至ることの重要性を主張している [Shneiderman 77]。著者はこの主張に基づいて、プログラミングにおいて有意義な一連のコードを部品として認識・習得し、複数の部品を組み合わせたより大きな部品を認識・習得、そして、それらの部品を組み合わせられる状態にあることを「構造的に理解」している状態と位置付けた。

そこで本章では、プログラミングにおける構造的理解を目指した「部品の段階的拡張手法」の提案 [古池 16] と、この提案手法を用いた学習支援システムの開発 [古池 17, 古池 18a] を行った。また、開発したシステムを用いた学習と、従来の教科書による学習、さらに開発システムからフィードバック機能を削除したシステムによる学習をそれぞれ比較することで、学習効果の評価を行った [古池 19a]。

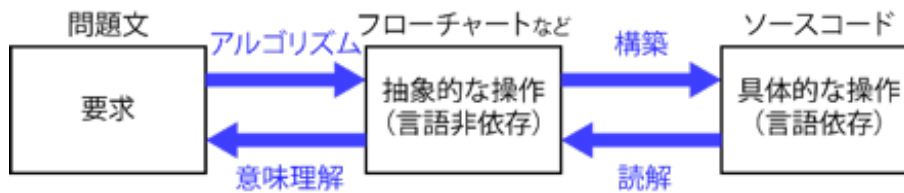


図 3.1 金森らによるプログラミングプロセス [金森 14]

3.3 先行研究

学習者は知識によって各行を理解することはできるものの、複数行になった際に理解が困難になることから、学習者がコードの各行についての関係性を正しく理解することを目的とした研究が行われている [東本 16, 東本 18, 渡辺 15]. 渡辺らの研究 [渡辺 15] では、金森らが提案するプログラミングプロセス [金森 14] (図 3.1) の中で、プログラムの読解と意味理解のプロセスに着目した. 渡辺らは、プログラムの読解と意味理解を行うことが学習において有意義であるとし、その手段として段階的抽象化プロセスを提案している. 段階的抽象化プロセスは、新開ら [新開 08] の段階的詳細化プロセスを参考に提案されている. 段階的詳細化プロセスとは、目的からプログラムを徐々に細分化していく手法で、アルゴリズム学習の支援を指向している. 一方で、段階的抽象化プロセスは、与えられたコードの中で一連の操作だと考えられる部分をまとめ、その意味を考えることを繰り返すことで、“段階的に読む”学習を支援するプロセスである. 例えば、(1)c に a を代入、(2)a に b を代入、(3)b に c を代入という 3 工程からなるプログラムでは、行ごとに見れば単なる代入の繰り返しであるが、3 工程をまとめて意味を考えることで、スワップという新しい概念を発見できるとしている. 著者は、この渡辺らの段階的抽象化プロセスによって、コードの各行の関係性を正しく理解することができ、有意義な塊としての認識が行えると考え.

しかし、新開らによる段階的詳細化プロセスや、渡辺らによる段階的抽象化プロセスでは、プログラムを構成している部品について着目しているものの、部品同士を組み合わせより大きな部品を作るといった、部品の発展については扱っていない. そこで著者は、コードの有意義な塊ごとに部品として認識すると同時に、その部品を段階的に発展させ組み合わせることによって、どうすれば部品を再利用できるかなどを考える“段階的に作

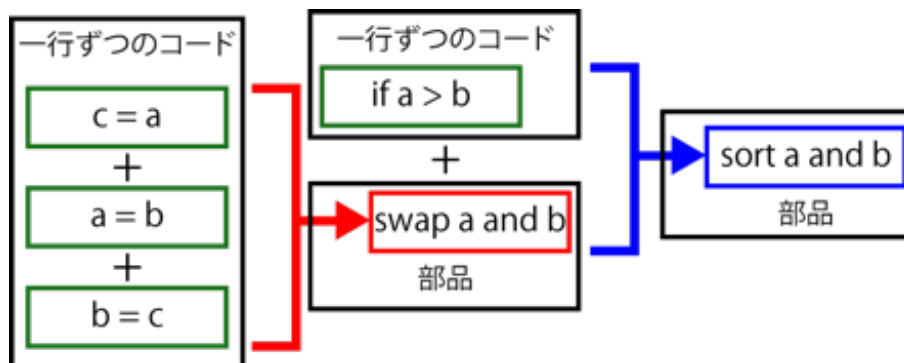


図 3.2 提案する「部品の段階的拡張手法」

る”学習に焦点を当てた。そうすることで、塊ごとの構造的な理解が深まるのではないかと考えた。

3.4 提案手法

著者は、プログラミング学習において知識の拡充のみでなく、構造的な理解も重要であると考えている。構造的な理解とは、知識の関係を正しく関係付けるためにコードを有意義な塊ごとに部品として認識し、その部品を用いてより大きな部品を構築できるといった理解である。よって、この構造的な理解を支援するための学習手法を提案する。

提案手法では、渡辺らの段階的抽象化プロセスを用いて、まず、学習者がプログラムにおける1行ずつのコードから有意な塊ごとに部品の構築を行う。さらに構築した部品に対して新たな処理や既存の部品を追加することで部品を変化させる。この過程を、著者は「部品の段階的拡張手法」と定義した。

こうした部品の変化を繰り返すことで部品の構造を段階的に拡張していき、学習者に部品ごとの関係性の理解を促し、構造的に学習することができる（図 3.2）。さらに、部品を段階的に拡張する行為そのものによって、自身で大きな部品を作る訓練にも繋がると考えられる。

3.5 提案システム

本章では、上述の「部品の段階的拡張手法」を用いた学習支援システム（以下、本システム）の提案を行う。本システムでは、各部品をブロックと位置付け、従来のプログラミングにおける1行ずつの処理をスタンダードブロックとし、スタンダードブロックに対して新たな処理や既存の部品を追加して拡張した部品をアドバンスブロックと定義した。このような単一の処理や複数の処理をブロックとして捉え、文法に依存しない内容そのものに焦点を当てる取り組みは広く行われている [Resnick 09, 松澤 14]。このようなブロック型プログラミングの考え方を導入することで文法的知識の概念を軽減しつつ、本章では内容そのものを段階的に発展させることを狙っている。

「部品の段階的拡張手法」の活用として、構築すべきアドバンスブロックを目標としてあらかじめ設定し、ブロックの組み合わせや発想が容易なものから順に習得を目指すことで、学習者におけるプログラムの構造的な理解を指向した、プログラミング学習の支援を図る。

なお、本システムはC#にて開発を行い、Windows10のデスクトップアプリケーションとして動作する。

3.5.1 ブロック

本項では、スタンダードブロックとアドバンスブロックからなる2種類のブロックの特徴を説明する。

ブロック全体が持つ性質として、「□=□」など記述のテンプレートと空欄が提供され、空欄には図3.3のシステムインタフェースにおけるブロックリストのように、初期値が既に記入された状態で提示される。学習者は問題文に合わせて、この記入欄を書き換えることが要求される。

スタンダードブロック

スタンダードブロックは、従来のプログラミングにおける各行に相当するものとし、「代入」や、「条件」、「繰り返し」などの基本的な処理を行うものと定義した。また、学習者

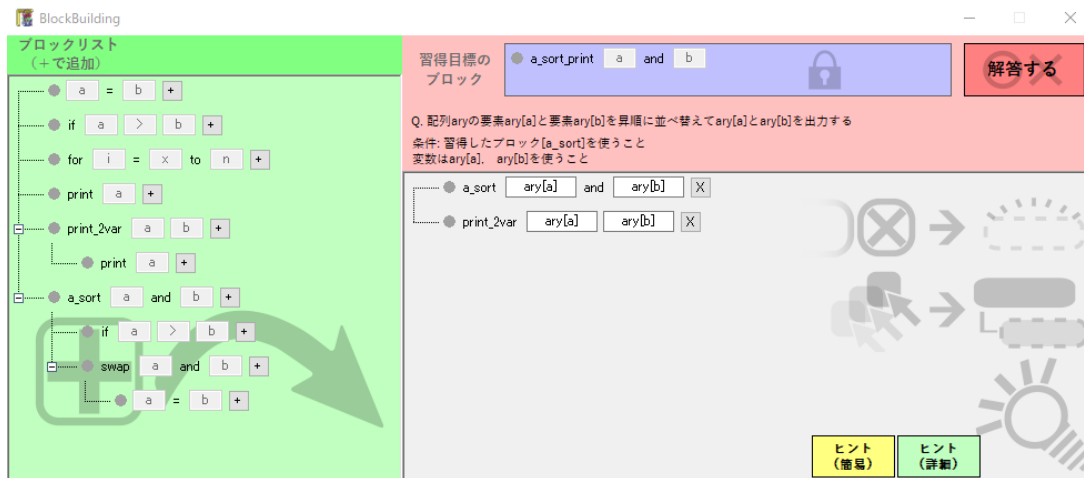


図 3.3 本システムのインタフェース

による構築に伴う負担を軽減するために、一般的な条件分岐などにみられる” ”などの表現はシステム上のインデントによる入れ子構造によって省略し,” else” については一つのスタンダードブロックとした。なお、これにより if を伴わない else の利用なども想定されるが、このような誤りは後述の正誤判断やフィードバック機能により修正することとなる。” (a == b) && (x == y)” などの複合条件に関しては、一つのスタンダードブロックとしては取り扱わず、and 条件であれば「条件」ブロックを2つ入れ子にして構成するものとし、or 条件であれば” else” と” if” の組み合わせで対応するものとした。

アドバンスブロック

アドバンスブロックは、従来のプログラミングにおける関数やクラスに相当するものとし、スタンダードブロックの組み合わせによる複合的な処理、すなわちアルゴリズム化された一意な部品と定義した。組み合わせの手段は三種類あり、まず、(1) スタンダードブロックのみの組み合わせによるアルゴリズムの定義、次に、(2) スタンダードブロックとアドバンスブロックの組み合わせによるアルゴリズムの拡張、加えて、(3) アドバンスブロック同士の組み合わせによるアルゴリズムの統合からなる。本システムでは、この三種類の組み立て方法を提供し、学習者はその中から選択、構築することで構造的理解の向上を目指す。

3.5.2 インタフェースと学習方法

本システムのインタフェースを図3.3に示す。本システムのインタフェースは、習得目標であるアドバンスブロックを習得するための習得画面のみからなる。まず、画面左側には、学習者が構築に利用できるブロックがブロックリストとして提示される。提示されるブロックは、用意されたスタンダードブロックの全てと、今までに学習したアドバンスブロックである。スタンダードブロックはそのまま一行で表示されるが、アドバンスブロックは階層構造を持って表示され、子ノードには包含するブロックが表示される。これらのブロックはカーソルを合わせることでそのブロックの動作が表示されるようになっている。画面右側ではそれらのブロックを用いた構築領域が提示され、ブロックリストから追加したブロックはこの構築領域に表示される。画面中央上部には習得するアドバンスブロックが提示され、完成した場合には、画面右上の解答ボタンを押すことで正解であれば次の習得画面に移行し、不正解である場合には不正解と表示される。この際、解答の正誤判定は、問題一つに対して正答例を一つ用意し、その他の解答は不正解とした。また、画面右下の簡易ヒントボタンか詳細ヒントボタンを押すことでフィードバックが提示される。

本システムは文法の評価機能やプログラムそのものの実行機能は持たず、インタフェースと各課題で設定された正解と学習者の解答を照合する正誤判定機能、正誤判定に基づくフィードバック機能から成り立つ。

3.5.3 段階的な発展

本システムでは、段階的な発展を行うために問題の前後関係を配慮してデザインされている。具体的には、ある問題は前の問題を包含する（前の問題のまとまりを再利用して解ける）ものに限定している。例えば、スワップを習得した後（図3.4）に、そのスワップを用いて2変数のソートを習得する（図3.5）。また、その後には2変数のソートを用いて、単純ソートを習得する。このように、本システムでは問題を段階的に発展させることで、学習者への構造的理解を促すことを指向した。

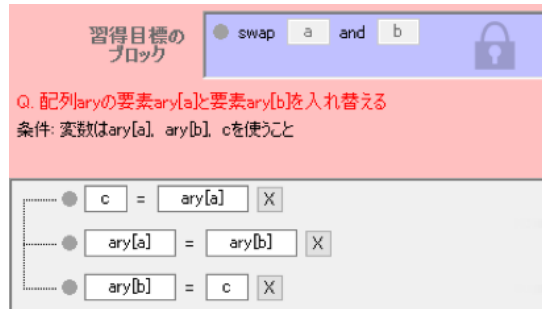


図 3.4 スワップの習得画面（一部）

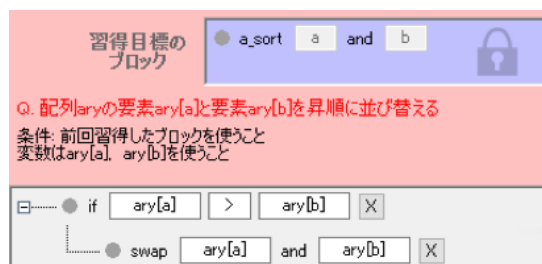


図 3.5 2変数ソートの習得画面（一部）

3.5.4 フィードバック

本システムでは、提案手法を活用したインタフェースで学習することが最も重要であると考えている。一方で、少なからず、不慣れであることや知識不足が原因で行き詰まる学習者が想定される。こういった学習者のために、本システムはフィードバック機能を備えることで行き詰まりの解消手段を提供している。

上述の簡易ヒントボタンと詳細ヒントボタンを押すことでフィードバックメッセージが提示される。システムは、システムにあらかじめ教材作成者が入力した正しいブロック構成と、学習者の作成したブロック構成を比較し、その差分箇所を誤り箇所としてフィードバックメッセージの形で提示する。フィードバックメッセージの優先順位は上から、「ブロックの数」、「階層の深さ」、「ブロックの種類」、「ブロック内における値」の順番となっている。簡易ヒントボタンでは、最も上流にあたる誤りに対して「～が間違っています」というフィードバックを行う。例えば、ブロックの数と階層の深さが合っており、ブロックの種類が違った場合には「上から n 番目のブロックの種類が違います」というフィードバックを行う。詳細ヒントボタンでは、最も上流にあたる誤りに対して、その正答を一つ

フィードバックとして提示する。例えば、ブロックの値が間違っていた場合は「上から n 番目のブロックの m 個目の値は x です」といったフィードバックを行う。

本システムではフィードバックの一つとして、ブロックの復習を導入している。具体的には、ブロックリストからアドバンスブロックの追加が行われた際に、追加されたアドバンスブロックの習得画面へ移行する。そこで、既に学習したアドバンスブロックを同様に構築することで、学習者の理解を深められると考えられる。追加されたアドバンスブロックの習得が終わった際には、元の習得画面に戻る。元の習得画面で一度そのブロックの復習を行った場合は、同一のアドバンスブロックの習得画面の間は再度復習画面が出ないようになっている。

3.6 評価実験 I

本システムの利用による学習者への学習効果を検討するために、構造化の作業を要する本システムと、構造化の作業が行われていない一般的なプログラミングの教科書を用いて評価実験を行った。

3.6.1 実験概要

実験参加者はプログラミングの講義で3年間プログラミングを学んだ大学3年生と4年生の計17名であり、for文、if文などの基本的な概念や、ソートなどのアルゴリズム、関数の学習については修了しているものである。また、本実験の対象言語はC言語をベースとしているが、本章ではプログラムの諸概念の理解には焦点を当てていないため、変数の定義やポインタの利用（あるいは参照渡し）などの正しさは要求しない。

事前に、基礎的なプログラミング能力や設計能力を測るための系列的に発展が可能な基礎課題4問、転移課題4問の計8問を記載した15分間の事前テストを行った。問題については表3.1に示すように、基礎課題では学習教材で学んだ範囲が定着しているかを確認するために、学習した範囲の問題を用意した。転移課題では、学習した内容が応用できるかを確認することを目的にしている。この研究では、既に学習したものを一連のまとまりとして習得し、そのまとまりを利用できる能力の獲得を目指しているため、差異の大きい

転移課題では、そもそも習得した内容を直接再利用することができない。そのため、学習範囲で学んだ内容を徐々に拡張し応用できるような比較的近い範囲で転移可能な問題を用意した。加えて、事前テストでは実験参加者に対して、なるべく各問でコンポーネント化などの構造化を行い、他の問題で再利用するよう指示を与えた。

表 3.1 評価実験 I: 基礎課題と転移課題のリスト

基礎課題
(1) 変数 a と変数 b を入れ替えるプログラムを記述せよ
(2) 変数 a と変数 b を昇順 (小さい数から大きい数) に並び替えるプログラムを記述せよ
(3) 要素数 n の配列 ary を昇順 (小さい数から大きい数) に並び替えるプログラムを記述せよ (ただしアルゴリズムは単純ソートとする)
(4) 要素数 n の配列 ary を昇順 (小さい数から大きい数) に並び替えるプログラムを記述せよ (ただしアルゴリズムはバブルソートとする)
転移課題
(5) 変数 a と変数 b, 変数 c を昇順 (小さい数から大きい数) に並び替えるプログラムを記述せよ
(6) 変数 a と変数 b を降順 (大きい数から小さい数) に並び替えるプログラムを記述せよ
(7) 変数 x が 0 のとき昇順 (小さい数から大きい数) に, 1 のとき降順 (大きい数から小さい数) に変数 a と変数 b を並び替えるプログラムを記述せよ
(8) 変数 x が 0 のとき昇順 (小さい数から大きい数) に, 1 のとき降順 (大きい数から小さい数) に要素数 n の配列 ary を並び替えるプログラムを記述せよ

各課題の評価は、(1) 正しいプログラムを書けるようになったかを評価するための手順点と、(2) プログラムを部品化し、利用できるようになったかを評価するための構造化点で採点することで行った。よって採点基準については図 3.6 に例示するように、(1) 手順点では、実験参加者が解答したプログラムの動作が問題文の要求を満たしているか否かで評価しており、(2) 構造化点では、他の問題で学習した一連のまとまりを再利用して、現在の問題を解いているか否かで評価している (なお、本章では文法の学習は対象としておらず、文法的な正しさは問わない)。なお、構造化点は、既学習の内容が存在しない基礎課題の間 1 のみ採点対象とならない。

事前テストの結果をもとに、実験群 9 名、統制群 8 名に振り分けて実験を行った。本実

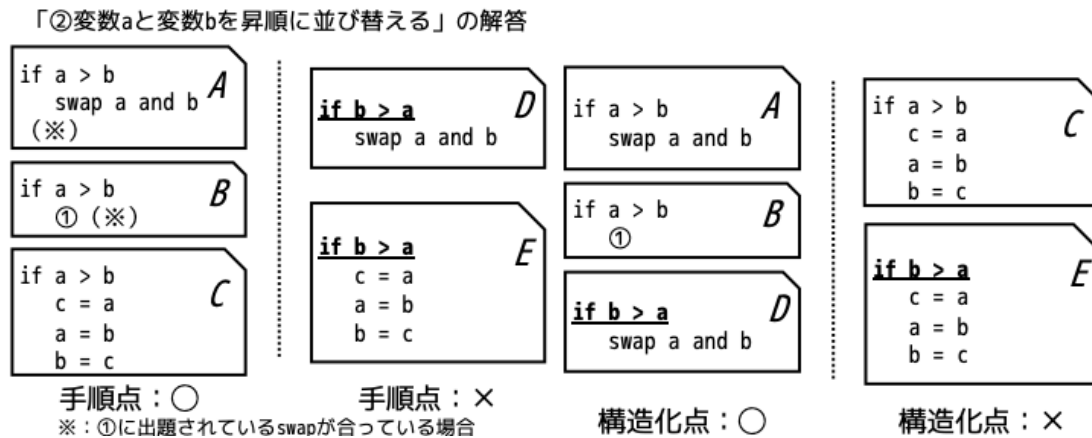


図 3.6 手順点・構造化点の採点基準の例

験では、まず実験群は 30 分間システムを用いて学習を行った。一方で、統制群は 30 分間一般的なプログラミングの教科書 [森元 07] を用いた学習を行った。その後、両群共に事前テストと同一の内容、評価で 15 分間の事後テストを行った。

3.6.2 テスト結果と考察

事前・事後テストについて、手順点については、各問 1 点で基礎課題が 4 問の 4 点満点、転移課題が 4 問の 4 点満点で合計 8 点満点となるように、構造化点については基礎課題の問 1 のみ再利用する課題がなく評価できない関係から、問 1 を除いて基礎課題が 3 点満点、転移課題が 4 点満点で合計 7 点満点となるように評価を行った。

まず、図 3.7 に手順点評価における事前・事後テストの平均値と分散分析の結果、その有意差 (* : $p < 0.05$) を示す。分散分析より、基礎課題 (図 3.7 左)・転移課題 (図 3.7 右) 共に群内で有意な差が見られた。

次に、図 3.8 に構造化点評価における事前・事後テストの平均値と分散分析の結果、その有意差 (* : $p < 0.05$) を示す。分散分析より、基礎課題 (図 3.8 左)・転移課題 (図 3.8 右) 共に群間、群内、交互作用に有意な差が見られた。基礎課題では、群間の事後テストの結果と群内の実験群に有意な差が見られた。転移課題でも同様に、群間の事後テストの結果と群内の実験群に有意な差が見られた。このことから、学習範囲で学習を行った基礎課題だけでなく、学習範囲で学ばなかった転移課題にも有意な差が見られた。

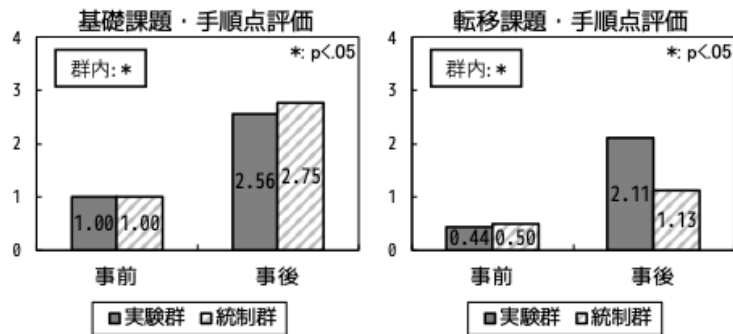


図 3.7 評価実験 I: 手順点における基礎・転移課題の評価

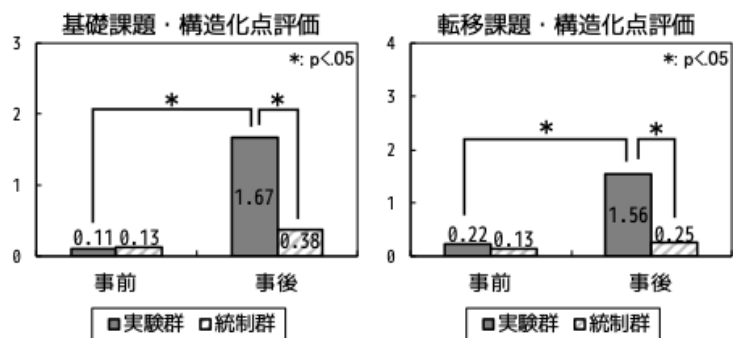


図 3.8 評価実験 I: 構造化点における基礎・転移課題の評価

以上の結果から、まず、実験群について考察する。実験群においては、システムによる課題提示やフィードバックが学習した範囲におけるプログラムの手順の理解と、コンポーネント化などによる再利用や部品を用いたプログラミングへの理解の向上および比較的近い範囲での応用を促進することがわかった。以上より、本システムがプログラムの構造的理解に寄与する可能性が示唆された。

次に統制群について考察する。統制群においては、一般的な教科書を用いることで、プログラムの手順の理解が認められたと考えられる。一方で、従来の学習ではコンポーネント化などによる再利用や部品を用いたプログラミングへの理解の向上が望めないと考えられる。

よって、システム利用学習と従来の学習における構造化点評価の結果から、システム利用学習において有意な結果が得られたため、システムが従来の学習と比較して構造的理解に有用であると考えられる。

一方、実験の結果から考察されたシステム利用における問題点を以下に指摘する。実験

参加者のシステム操作ログについて分析したところ、本システムで扱う正答例が講義資料や教科書、参考書などに登場する典型的なものであったためか、殆どの学習者は正答例のように組み立てていた。また、そうでない学習者も、ヒント機能を用いたフィードバックにより正答例に近づくように修正が行われていたことが観察できた。しかし、今回システム上で扱った正答例が1つであったために、学習者から別解が発生することは十分に予想される。よって、今後は本システムで扱っている正解があくまで正答例の1つであることを学習者に伝えた上で、システムにおける正答例を発見するよう指示する必要があると考える。

3.7 評価実験 II

評価実験 I では、本システムに学習効果があることを検証し、紙教材学習との比較において構造的理解に対して有意な学習効果が得られている。しかし、この学習効果が学習支援システムのフィードバックによる効果であるのか、あるいは提案手法そのものが持つ効果であるのかについては議論できていない。つまり、提案手法を用いた構造化の作業が有効であったのか、もしくはシステムが行ったフィードバックが有効であったのかは明らかではない。そのため、フィードバック機能のある従来の本システムを用いた学習と、本システムからフィードバック機能を削除したシステムを用いた学習の比較実験を行うことで、提案手法とフィードバック機能の有効性をそれぞれ調査する必要があると考える。よって本実験では、システムにおいてフィードバックの有無による学習効果の違いを調査した。

3.7.1 実験概要

実験参加者は評価実験 I と異なる実験参加者かつ、プログラミングの講義で3年間プログラミングを学んだ大学生15名であり、for文、if文など基本的な概念や、配列の並び替えなどのアルゴリズム、関数の学習については修了しているものである。また、本実験の対象言語はC言語を基にポインタや関数マクロの概念を省略したものとなっている。

事前では、評価実験 I と同様に系列的に発展が可能な問題を記載したテストを2種類

実施した。なお、評価実験 I で行ったテストから、問題間の難易度の飛躍があったことやソートの範囲の部品に限定されていたことの改善を試みて、基礎・転移課題に対して新たに問題を追加した。よって、テストはそれぞれ、学習範囲の問題 11 問を基礎課題（表 3.2）として 10 分間、学習範囲から転移が可能な問題 9 問を転移課題（表 3.3）として 10 分間で行った。テストを行う際には、各問でコンポーネント化など構造化を行い他の問題やテストで再利用するよう指示を与えた。

表 3.2 評価実験 II: 基礎課題のリスト

基礎課題
(1) 変数 a と変数 b を入れ替えるプログラムを記述せよ
(2) 変数 a と変数 b を出力するプログラムを記述せよ
(3) 変数 a と変数 b を入れ替えて出力するプログラムを記述せよ
(4) 変数 a と変数 b を昇順（小さい数から大きい数）に並び替えるプログラムを記述せよ
(5) 変数 a と変数 b を昇順（小さい数から大きい数）に並び替えて出力するプログラムを記述せよ
(6) 要素数 n の配列 ary を出力するプログラムを記述せよ
(7) 配列 ary の i から要素数 n までの最小値を ary[i] に求めるプログラムを記述せよ
(8) 配列 ary の i から要素数 n までの最小値を ary[i] に求めて出力するプログラムを記述せよ
(9) 要素数 n の配列 ary を昇順（小さい数から大きい数）に並び替えるプログラムを記述せよ（ただしアルゴリズムは単純ソートとする）
(10) 要素数 n の配列 ary を昇順（小さい数から大きい数）に並び替えて出力するプログラムを記述せよ（ただしアルゴリズムは単純ソートとする）
(11) 要素数 n の配列 ary を昇順（小さい数から大きい数）に並び替えて出力するプログラムを記述せよ（ただしアルゴリズムはバブルソートとする）

これらの評価は評価実験 I と同様に手順が正しいかを評価した手順点と、コンポーネント化など構造化を行い他の問題で再利用できているかを評価した構造化点の 2 種類にて 1 問 1 点として評価した。

事前テストの後、学習教材としてフィードバックありの既存システムを使う FB 有群 8 名と、既存システムからフィードバック機能を削除したシステムを使う FB 無群 7 名に分かれて 30 分間学習を行った。

事後では事前と同じテストを実施し、事前と同様の評価基準で評価した。さらに、実験

表 3.3 評価実験 II: 転移課題のリスト

転移課題
(1) 変数 a と変数 b, 変数 c を昇順 (小さい数から大きい数) に並び替えるプログラムを記述せよ
(2) 変数 a と変数 b, 変数 c を出力するプログラムを記述せよ
(3) 変数 a と変数 b, 変数 c を昇順 (小さい数から大きい数) に並び替えて出力するプログラムを記述せよ
(4) 変数 a と変数 b を降順 (大きい数から小さい数) に並び替えるプログラムを記述せよ
(5) 変数 a と変数 b を降順 (大きい数から小さい数) に並び替えて出力するプログラムを記述せよ
(6) 変数 x が 0 のとき昇順 (小さい数から大きい数) に, 1 のとき降順 (大きい数から小さい数) に変数 a と変数 b を並び替えるプログラムを記述せよ
(7) 変数 x が 0 のとき昇順 (小さい数から大きい数) に, 1 のとき降順 (大きい数から小さい数) に変数 a と変数 b を並び替えて出力するプログラムを記述せよ
(8) 変数 x が 0 のとき昇順 (小さい数から大きい数) に, 1 のとき降順 (大きい数から小さい数) に要素数 n の配列 ary を並び替えるプログラムを記述せよ
(9) 変数 x が 0 のとき昇順 (小さい数から大きい数) に, 1 のとき降順 (大きい数から小さい数) に要素数 n の配列 ary を並び替えて出力するプログラムを記述せよ

終了後に実験参加者にアンケートやインタビューを行った。

3.7.2 テスト結果と考察

事前・事後テストについて、手順点については、各問 1 点で基礎課題が 11 問の 11 点満点、転移課題が 9 問の 9 点満点で合計 20 点満点となるように、構造化点については評価実験 I と同様に基礎課題の問 1 のみ再利用する課題がなく評価できない関係から、問 1 を除いて基礎課題が 10 点満点、転移課題が 9 点満点で合計 19 点満点となるように評価を行った。

まず、図 3.9 に手順点評価における事前・事後テストの平均値と分散分析の結果、その有意差 ($* : p < 0.05$) を示す。分散分析より、基礎課題 (図 3.9 左)・転移課題 (図 3.9

右)において群内で有意な差が見られ、基礎課題においては交互作用が見られた。両課題においてどちらの群にも有意に得点が向上しており、さらに、事後の基礎課題においてはFB有群がFB無群よりも有意に得点が高かった。よって、提案手法が有効だったものの、学習範囲においてはフィードバックにより正解を知ることにより効果的に学習ができたと考えられる。

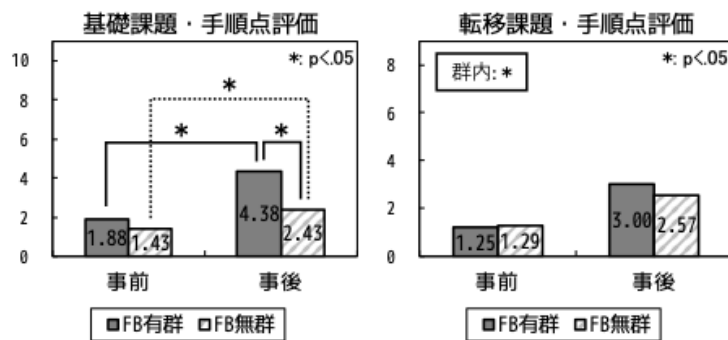


図 3.9 評価実験 II: 手順点における基礎・転移課題の評価

次に、図3.10に構造化点評価における事前・事後テストの平均値と分散分析の結果、その有意差 ($*: p < 0.05$) を示す。分散分析より、基礎課題 (図3.10左)・転移課題 (図3.10右)において群内に有意な差が見られ、基礎課題においては交互作用が見られた。さらに、基礎課題においてFB有群のみに有意な差がみられた。よって、手順点評価における結果と同様に、学習範囲においてはフィードバックにより正解を知ることにより効果的な学習ができたと考えられる。さらに、転移課題では、両群共に有意に得点が向上していたが、群間に差はみられなかった。このことから、提案手法自体が有効であったと考えられる。

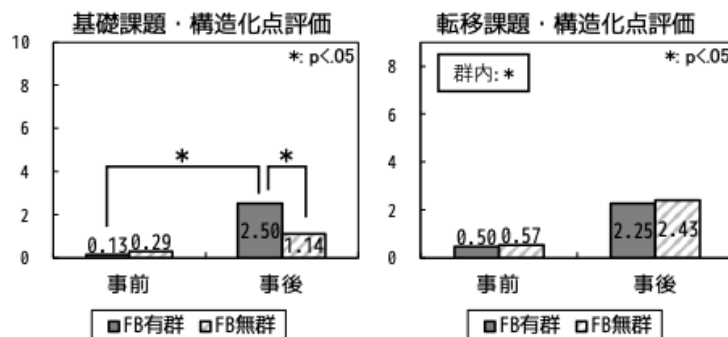


図 3.10 評価実験 II: 構造化点における基礎・転移課題の評価

以上の結果をまとめると、学習範囲においてはフィードバックにより正解を知ることにより効果的に学習できたと考えられる。

本提案手法は、既に学習したものを一連のまとまりとして獲得して、それを利用できる能力の獲得を目指している。これの評価を意図したものが構造化点であり、評価実験 I では本システムが紙教材での学習より構造的理解に効果があることを示し、評価実験 II では、学習範囲において本システムのフィードバックが構造的理解に効果があることが認められ、学習範囲におけるフィードバックの有効性を示し、応用範囲においてはフィードバックがないシステムでも一定の学習効果が認められ、本提案手法そのものが構造的理解に寄与することを示した。一方で、フィードバックがないシステムを利用した群と、紙教材で学習した群の直接的な比較は行っていないため、構造化作業を要求する提案手法そのものが紙教材学習より有効であったのかは示せていない。

3.7.3 評価実験 I との統合的な分析

紙教材を用いて学習した群と、フィードバックがないシステムを用いた群、およびフィードバックがあるシステムを用いた群を直接比較するために、評価実験 I と評価実験 II に共通する問題についてのみの点数を集計し、分散分析を行った。システムフィードバック有群 (FB 有群) が 17 名、システムフィードバック無群 (FB 無群) が 7 名、紙教材学習群 (紙教材群) が 8 名の 3 群に分けて、事前事後を要因とした分析を行う。

図 3.11 に構造化点評価における事前・事後テスト、分散分析の結果を示す。図 3.11 から、転移課題においては FB 有群と FB 無群の事後にのみ有意に向上していた。基礎課題については、FB 有群の事後にのみ有意に向上していた。

以上の結果は評価実験 II の結果を踏襲しており、応用的な範囲においてはフィードバックの有無に関わらず提案手法が構造的理解に寄与したが、紙教材では有意な差は得られなかった。一方、学習範囲については、フィードバック機能そのものが学習範囲の定着に寄与していると考えられる。

ただし、今回の分析については、各群の実験参加者数にばらつきがあり、また、分析した問題自体は評価実験 I, II で共通した問題であるが、評価実験 II では出題する問題数を増加させた関係で、途中で他の問題をはさんでいるため、今後は実験条件をコントロール

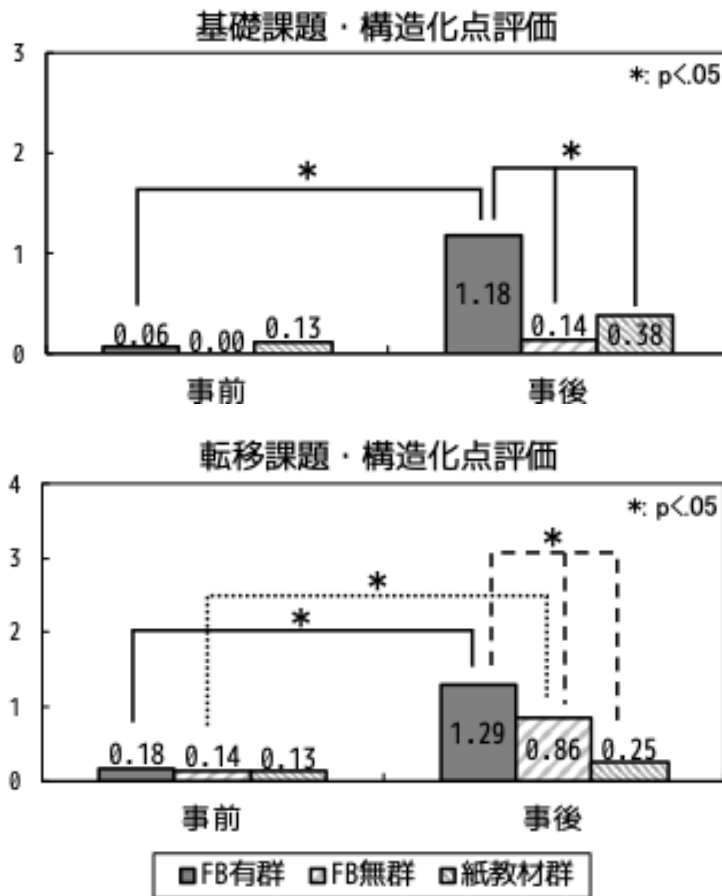


図 3.11 統合的な分析: 構造化における基礎・転移課題の3群の比較評価

した上で、より厳密な実験を行う必要がある。

3.7.4 アンケート・インタビュー結果と考察

両群に行ったアンケートの結果を表3.4に示す。表3.4では、4件法のアンケートのうち3と4の評価をポジティブな回答として集計した。表3.4から、両群共にシステム学習がプログラムの塊を部品として認識することにつながると感じており、さらに実験を通してプログラムを構造的に理解することを重要だと考えていることがわかった。このことから、提案手法が学習者に受け入れられていると考えられる。一方で、フィードバックありのシステムを利用した群は全ての実験参加者が、システム学習はプログラムの理解や、プログラムごとの再利用性を高めるプログラミングにつながると感じているのに対して、

フィードバック無しのシステムを利用した群は感じないと回答した実験参加者が半数程度存在した。このことから、フィードバックがあることにより再利用することの効果を実感できていたと考えられる。

表 3.4 アンケート項目とポジティブな回答数の比較

質問項目	FB 有群 (8 名)	FB 無群 (7 名)
(システム学習が) プログラムの塊を部品として認識することにつながると思うか	8	6
実験を通してプログラムを構造的に理解することは重要だと思ったか	8	6
(システム学習が) プログラミングの理解につながると思うか	8	4
(システム学習が) プログラムごとの再利用性を高めるプログラミングにつながると思うか	8	4

実験後のインタビューから、実際に FB 無群の複数の実験参加者が「フィードバックが無いシステムでも、教科書より有用だった」や、「システムを使うことでプログラムの見方が変わった」とコメントしている。このことから、提案手法自体が構造的理解にある程度有効であったと考えられる。

3.8 関連研究と議論

本節では、本章を位置付け、よりその価値を明らかにするために周辺の研究について報告し、本章との差異を述べる。

江木ら [江木 09] は、プログラミング教育においては、プログラミング言語、アルゴリズム、構築技術の 3 つの主要な側面があるとしたうえで、学習者にとってはこれらの 3 つは不可分であり、ほぼ同時進行で学習が進むことを指摘している。その上で、多くの研究がプログラム言語やアルゴリズムの学習に焦点が置かれており、構築技術についての教育支援（特に初心者向けのもの）が不足していることを指摘している。本章はこの位置付けにおいては、アルゴリズムと構築技術の双方に焦点を当てたものになる。アルゴリズム的な観点からは、ある一連のコード（部品）の持つ機能的な意味と、その一連のコード（部

品)を拡張あるいは修正したあとのコード(拡張した部品)の関係性の理解を支援している。そのため、アルゴリズム的な側面からは、コードの差分が持つ機能的な差の獲得につながる事が期待できる。さらに、構築技術の観点からは、一連のコードを部品としてまとめ、それを再利用することで、既存の知識を基にした設計のスキルの育成につながる事が期待できる。本章では、既存の知識を基にした再利用をベースとした設計に対して評価を行い、通常の教科書ベースの学習と比較しての効果を検証した。

江木ら [江木 09] や東本ら [東本 16, 東本 18] はトレースを支援する方法、山本ら [山本 18] はトレースを包含する体系的デバッグ手順を指導する方法を提案している。しかしながら、作成したコードの拡張や発展といった観点での構築技術については言及していない。一連のコードをまとまりとして捉え、その機能的な意味を理解するときにデバッグ(トレース)は重要である。このような再利用を繰り返すことで、自身が使える知識としての定着を図る本章のアプローチも重要であると考えられる。

構築技術に関しては、Kölling ら [Kölling 03] の BlueJ と呼ばれるオブジェクト指向の理解のための学習支援システムも有名である。本システムは、オブジェクト指向を初学者が学ぶために必要な要素に限定し、教育を支援する意図で開発されている。しかしながら、オブジェクトファースト(つまり、オブジェクトを理解させてからコードを理解させる)方法は、車の下にスポーツカー、果物の下にリンゴという概念が存在するなどの、学習者にとって既知の構造であればオブジェクトをもとに理解することが容易になる可能性があるが、全てのプログラムが学習者にとって最初から理解可能な有意義な塊ではなく、コードドリブンで一連のコードの理解し、まとまりを大きくするという本章のアプローチもまた必要であろうと考える。

一連のコードとその振舞い、あるいは機能を関連付けることを指向した研究も存在する [山本 14, 杉浦 08, Nakahara 09]。山本らの研究 [山本 14] では、学習者の入力したプログラムと正解のプログラムの振舞いの差を対象世界の差として可視化している。しかし、この振舞いがどのような機能を持つか、さらに一連のコードを再利用し、発展させるという点には着目していない。杉浦ら [杉浦 08] や Noguchi ら [Nakahara 09] の研究においても、あるアルゴリズムを理解するために、その振舞いを学習者自身が再現するという学習方法が提案されているが、やはり再利用や拡張という点は焦点とされていない。

松田ら [松田 97b, 松田 97a] は、再帰プログラミングの支援のための手法を提案している。これらは再帰プログラミングを対象として、詳細な構造を検討しており、再帰構造における構成要素を部品として捉え、部品を入れ替えたとしても、再帰の構造は変化しないという観点から、一連のコードとその一連のコードが示す再帰における機能の関係を理解を支援している。再帰構造という理解が困難な領域においてその構造を理解させることに特化しており重要な研究であるが、本章のように一連のコードを再利用したり、拡張するという点には着目しておらず、構造そのものを変化させる点については検討していない。

上記より、プログラミングの領域において、本章のようにボトムアップな思考での学習支援に関する研究は多くはない。よって再利用性や拡張といった観点に着目し、学習者に演習を要求し、部品的な理解を支援するという本提案システムは新規性を有すると考える。

3.9 本章のまとめ

本章では、プログラミング学習において、コードを有意味な塊ごとに部品として認識し、それらの部品を再利用し、知識を組み合わせるより大きな部品の構築を構造的に行えるように学習者の既有知識を整理するといった「構造的理解」の重要性を主張した。その上で、構造的理解の支援を行うために、先行研究の段階的抽象化を用いた「部品の段階的拡張手法」の提案を行った。部品の段階的拡張手法では、プログラムにおける一行ずつの処理を組み合わせる部品の構築を行い、さらに構築した部品に対して新たな処理や既存の部品の追加を行うことで、段階的に部品を拡張していくことが有用であると考えた。加えて、提案手法を支援するためのシステムの開発を行った。さらに、本システムを用いた評価実験を行った結果から、本システムは学習効果を有しており、従来の一般的な教科書を用いた学習よりも有効であったことや、システムのフィードバックの有無に関わらず提案手法が有効であったこと、システムのフィードバックによってシステムで学習する範囲においてはさらに学習が促進されたことが確認できた。これらの結果から、本システムが構造的理解に寄与するものと考えられる。

今後の課題としては、長期的なシステム利用による学習効果の検証や、システムフィードバックのフィードバック内容についての精査などが挙げられる。

第4章 機能・振舞い・構造の観点に基づく問題解決過程のモデル

4.1 本論文における本章の位置付け

3章では、領域知識モデルの構築に先駆けて、任意の知識粒度で取り扱うことができる教授モデルを提案し、その有効性を確認した。本章では、3章で取り扱った学習過程を考察することで、領域知識モデルの構築を行う。まず、3章において定義されている知識である部品（ソースコードと機能のペア）の定義を発展させ、より厳密な定義を与える。具体的には、2章で述べたように、プログラミングとは何か（情報）を操作して処理する機械システムであると考え、定性物理やオントロジー工学で議論されてきた人工物の概念化 [De Kleer 84, 西田 93, 笹島 96, 来村 02b, 来村 02a] に近似することで、新たに部品に振舞いの概念を導入する。これにより、3章で提案した学習活動の問題解決中に扱われる知識として「部品」を知識記述的に定義し、その部品の獲得や拡張における問題解決プロセスを考察することで学習者の学びを支援するための領域知識モデルを提案する [古池 20]。

4.2 プログラミング学習支援としての本章の動機

近年、社会の情報化に伴い IT 人材の不足が叫ばれ、プログラミング学習の需要が高まっている。さらには Computational Thinking [Wing 06] の獲得を目指して、小学校におけるプログラミング教育の必修化を筆頭に素養のためのプログラミング学習についても注目されている。よってこれに伴い、プログラミング学習における知的学習支援を高度化することの重要性も高まっていると著者らは考えている。

プログラミング学習で目指すべきことの一つに、知識の体系化がある。すなわち、学習者が適切なタイミングで既存の知識を再利用することができるように知識を整理することである。計算機がこのような学びを知的に支援するためには、個々の学びに対して適応的であることが望ましい。知的学習支援に関する研究では個々の学習手法に対して、(A) 出題する問題の特徴をインデクシング（特徴記述）により制御すること [平嶋 92, 平嶋 95, 東本 08] や、(B) 学習者の問題解決過程をモデルベースに把握して

フィードバックなどの介入を適切に行うこと [平嶋 92, 平嶋 95, 平嶋 99] が有効であることが知られている。

(A) については、ある問題の解決がその解決に要する知識の獲得を指向しているとみなせば、プログラミング知識の記述そのものが問題のインデクシングに繋がる。先行研究では、機能を発揮する実体を部品と呼び、機能 (function) – 振舞い (behavior) – 構造 (structure) の観点を導入し、さらにこの 3 要素を併せて、部品の知識として取り扱うことで知識記述に役立っている研究が多く存在する。その一つである定性物理においては、振舞いとは「対象の構造における因果性を持つ複数の入出力関係」であり、機能とは「振舞いに機能を表す語彙を割当てたもの」だとしている [De Kleer 84, 西田 93]。定性物理ではこの定義に基づき、部品の組み合わせや修正により現象や問題解決を定性的に表現することを可能としている。また、知的学習支援においても、学習対象の説明に機能–振舞い–構造の観点を導入した例は少なくない (e.g. [柏原 91, 松田 97a])。

オントロジー工学 [來村 02b, 來村 02a, 笹島 96] では「すべての人工物は部品の組み合わせで構成されている」という前提に立つことで、対象物を、入力を処理して利用者が要求するものに変換して出力する部品の合成としてみなしている。これにより、部品の構造をより粒度の小さい部品同士の関係性で表現することを可能にした。さらにオントロジー工学 [笹島 96] は定性物理の考えを発展させ、従来の機能と振舞いの考えに対して異なる定義を与えた。すなわち振舞いを「時間とともに変化する対象の状態遷移であり、利用者が必要とする対象のシミュレーションの結果」と定義し、機能を「その部品が組み込まれた系の目標に基づいて解釈した結果」と定義した。

オントロジー工学の考え方に則り、これらの 3 要素を用いることでプログラミングの問題解決過程 (図 4.1) についての説明が可能になる。まず、構造であるソースコードに対して、変数の初期状態を入力として与えることで終了状態が出力されるとすれば、この入力と出力の差分、すなわち構造がもたらした差分を構造の振舞いとして観察することができる。さらに、観察した振舞いについては、組み込まれた系の目標 (context) を想定して機能として意味づけて解釈することが可能である。

したがって、本章では機能を発揮する実体である部品を一定の機能を果たすソースコードの有意義なかたまりとみなすことによってオントロジー工学の考え方を導入し、機能–

振舞い-構造の3要素を併せて部品の知識を扱うことでプログラミングにおける知識を表現する。

(B) のプログラミングにおける問題解決過程は、古くからシステム工学の分野で Program Comprehension または Program Understanding として捉えようと試みられてきた。そこでは、数々のモデル (cf. [Harth 17, von Mayrhauser 95]) が提案されてきたものの、学習支援を指向したモデルは未だ提案されていない。

よって本章は、プログラミングにおける知識を部品の知識として定義することで、これまで暗黙知とされてきたプログラミングに関する諸要素に接近し、プログラミング言語独立に各要素を明らかにして整理した。これにより、学習の観点から部品の知識を用いた問題解決過程のモデルを構築し、暗黙知を形式知化することを試みた。また、このモデル構築を通して、これまで主に学習支援が行われてきたのは実世界の活動だけであり、学習者の表象での活動について支援が行われていない、という現状を指摘した。

本章のモデルに基づくことで、プログラミング学習の知的支援において問題の適応的な提示や正誤判定機能を実現するだけでなく、学習プロセス全体における誤り箇所の把握や、学習者の知識状態の推定、さらには試行錯誤を促す支援の実現も期待できる。

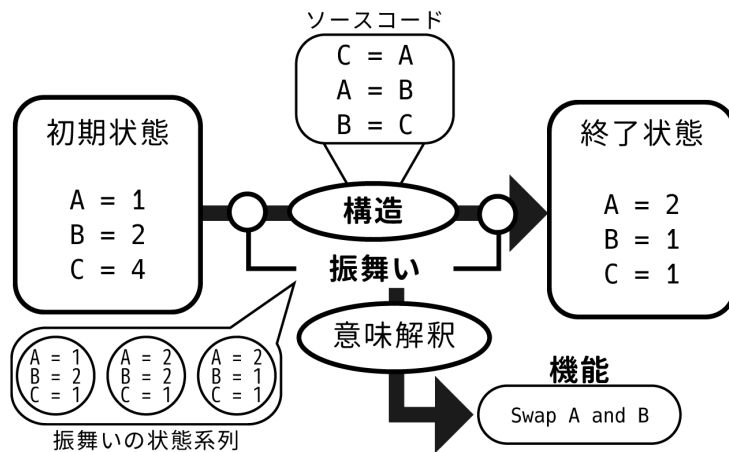


図 4.1 プログラミングにおける問題解決過程

4.3 プログラミングにおける知識の再利用性

本章では知識を部品の知識として取り扱うために、プログラミングにおける知識の再利用性について考察する。

プログラミングにおいては、要求が解決すべき問題となる。要求は、ソフトウェア工学領域において機能要求と非機能要求に分けて定義されている。機能要求は何を果たすかの要求であるのに対して、非機能要求はインタフェースやセキュリティなど、機能要求を満たすうえでの特性である。本章では、機能要求の達成をプログラミングにおける問題解決とみなして、この過程に関わる知識を取り扱うことを目的としている。よって本章で取り扱う知識は、プログラムを構成するソースファイルのサイズやステップ数等の知識は対象外であり、プログラミングにおける機能に関する知識のみを対象にしている。

通常、プログラミングで機能要求を満たすためには、その機能要求を満たすソースコード（構造）を構築して機能を実現する必要がある。例えば、システム開発業務の文脈では、「物流業務管理」など高度で複雑な機能要求がなされる。この要求では、「入庫出庫管理」や「輸送管理」、「梱包管理」などといった複数の機能を満たすことが要求に内包される（図4.2）。「物流業務管理」の機能を実現するとき、複数の機能が内包されていることを意識できれば、「A社の物流業務管理」だけでなく、「B社の梱包管理」にもA社の個別の機能を再利用することができる。より一般的には、プログラミングにおける知識の再利用性とは、学習者が機能と構造を関連付けて部品として表現される知識として獲得し、新たな問題解決において獲得済みの知識を再利用できることと本章では定義する。さらに、この知識は階層構造などで体系化されていることが再利用の観点で重要であり、特定の機能を達成するために獲得した構造はその階層性を理解している限り、必要なときに必要な粒度の階層で再利用することができる。したがって、システム開発者はこういった複数の機能が内包されていることを意識し、それぞれ独立の機能として扱えることである機能に関する知識を再利用することが可能になる。このとき、再利用されるものは一定の機能を実現する有意味なソースコードのかたまり、すなわち部品である。しかし再利用するためには、機能に関する知識も必要になる。この機能に関する知識を本章では、機能-振舞い-構造の3要素で表現する知識群だと考えている。この知識群を、「部品知識」と定義する。

さらに、プログラミングにおける機能に関する知識を部品知識として扱うことの妥当性

を裏付ける研究がいくつかある。Chase らは、チェスにおける初心者と熟達者の違いを調査し、プレイ中に想起される構造化された有意味な知識のかたまりの数に差があることを示した [Chase 73]。さらに Chase らは、有意味な知識のかたまりが体系的に組織されており、それぞれの有意味な知識のかたまり内の知識が機能的な関係性を持っていると主張した。また、Adelson は、Chase らの調査を受けてプログラミングにおける初心者と熟達者の違いを調査し、同様に想起されることを確認し、プログラミングにおいても有意味な知識のかたまりが体系的に組織されていることを示唆した [Adelson 81]。本章ではこの有意味な知識のかたまりが、先述した部品知識に相当すると考えている。したがって、プログラミングにおける知識の再利用性は構造化された部品知識によって担保されていると考えられる。

実際に、部品知識の獲得を指向して学習支援を行った事例も存在する。古池らの研究 [古池 18a, 古池 19a] では、本章と同様にプログラミングにおけるソースコードの有意味なかたまりを部品として取り扱うことで、学習者による部品知識の獲得を促進する学習方法である「部品の段階的拡張手法」を提案している。さらに、その学習方法を支援する学習支援システムを開発し、学習方法自体の有効性やシステムによる学習効果の向上を確認している。また古池らは、部品知識の一般化—特殊化や全体化—部分化の関係性に着目することで、漸進的な問題提示を目的とした学習課題のグラフ構築に関する議論を行っている [Koike 18b, Koike 18c]。しかし古池らの研究では、部品知識における機能—振舞い—構

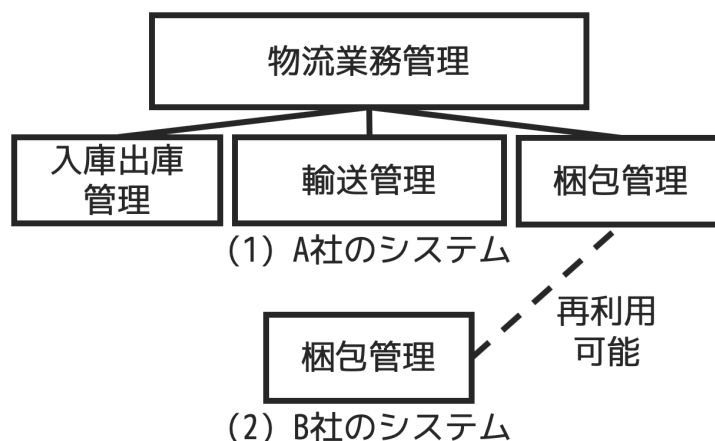


図 4.2 機能が再利用可能な例

造の区別，特に機能と振舞いの差は十分に議論されていない．したがって，本章では古池らの研究で行われていた議論を発展させる形で，部品知識における機能–振舞い–構造の区別を定義する．

4.2 で述べたように，ソフトウェア工学領域ではプログラミングの知識分類や問題解決過程についての研究が盛んである．Shneiderman ら [Shneiderman 79] は，プログラミングの知識を Semantic knowledge と Syntactic knowledge に大別して定義し，問題解決過程を説明している．Semantic knowledge は代入文で何が行われるか，バブルソートとは何か，などの機能的な意味概念の知識であり，Syntactic knowledge は代入文の書き方や for 文の書き方などの構文表現の知識である．Shneiderman らはこの 2 つの知識によって，システム開発者の問題解決がなされているとした．しかし，Semantic knowledge と Syntactic knowledge は経験則に基づいて分類されており，ここでの Semantic knowledge は機能と振舞いが混同したもので，Syntactic knowledge は構造だと受け取れる．そのため Shneiderman らの知識の分類では，問題解決過程における機能と振舞いの差が示せていない．von Mayrhauser ら [von Mayrhauser 95] や Harth ら [Harth 17] は，Shneiderman らのモデルも含めた代表的なプログラミングの問題解決モデルを概説して比較しているが，学習支援を指向したモデルはなく，またどのモデルも，知識において機能と振舞いの差を区別できていない．

部品として知識の再利用性を高めるためには，機能–振舞い–構造を明示的に分けて記述すべきであることが指摘されている [De Kleer 84, 笹島 96]．したがって，本章ではこの 3 要素を区別し，それらを併せて部品の知識として取り扱うことで再利用性の向上をねらう．

また，本章で扱う知識の範囲を明らかにするため，プログラミング学習における機能に関する知識を二つに大別した．一つは，機能–振舞い–構造の 3 要素を組み合わせた知識であり，本章において部品のモデルとして表現される知識である．もう一つは，部品の「構築方法」や，「獲得方法」，「拡張方法」などの部品の取扱いに関する知識である．以降では，前者をただの知識もしくは部品知識と呼び，特に後者をメタ部品知識と呼ぶ．本章では，問題解決過程を通して部品知識を獲得していく中で，メタ部品知識の獲得も指向しているものの，明示的には取り扱っていない．よって，本章では主に部品知識を議論の対象

としており、メタ部品知識については十分に議論できていない。これについては 4.7.2 にて詳細を述べる。

4.4 部品知識の獲得・拡張活動のプロセス

本章では、まず部品のモデルを定義し、問題解決過程を部品の活用・拡張プロセスであるとみなすことで、そこで行われる学習を部品知識の獲得・拡張活動プロセスと定義したモデルを提案する。したがって、部品の定義について 4.4.1 で、部品の構築がどのようにして行われるかを 4.4.2 でそれぞれ議論した後、4.4.3 にて、部品の定義と構築活動を踏まえて部品知識の獲得・拡張活動プロセスのモデルを提案する。

4.4.1 部品の定義

4.2 で述べた通り、本章における部品は、一定の機能を果たすソースコードの有意味なかたまりと定義している。この部品を再利用するために、機能-振舞い-構造の 3 要素を併せた部品知識が必要となる。本章では、この定義の詳細と部品知識について説明する。

まず最初に、本節では 4.2 にて導入した「機能」、「振舞い」、「構造」について改めて定義する。本章における「構造」とは、「ソースコードやフローチャートなどにみられる処理の順序関係や接続関係」と定義する。さらに、振舞いとは「部品が対象とする入出力の状態遷移」と定義する。このとき、入出力とは標準入出力や関数の引数だけでなく、ある一連の構造が処理を行う対象物の事前状態（入力）と事後状態（出力）と広義で定義する。さらに、機能を「部品がもたらす振舞い（入出力の状態遷移）を目標に基づいて解釈したもの」と定義する。そのため、本章における部品の定義は、「構造の有意味なかたまり」であるといえる。この有意味なかたまりとは、10 ステップ単位や関数単位などの定量的なものではなく、学習の観点から「人間が機能として有益である（再利用できる、学習する価値がある）と考えるもの」とし、「有意味」を機能として解釈できる単位と定義する。

通常、プログラミングにおいては機能要求を満たすソースコードを構築することが求められる。誤解されがちだが、熟達したシステム開発者は機能を達成するときには直接ソースコードを考えているわけではなく、ソースコードをどのような手順で実行し、どのよう

に振る舞えば機能を達成できるか、を思考しながらソースコードを構築している。このことは、Shneiderman らの研究において Semantic knowledge として「代入文で何が行われるか」が扱われることから明らかである [Shneiderman 79]。例えば、C 言語において `for(int i = 0; i < 0; i++)` と `for(int i = 0; i < 10; i++)` はソースコードとしてみればほとんど同じソースコードであるが、意味的には大きく異なる。これが異なるといえるのは、`i < 0` であるか、`i < 10` であるかによって for 文の振舞いが増えるからである。すなわち、熟達したシステム開発者が繰り返しの機能に対して for 文を用いることができるのは、for 文がどのように振る舞うかを知っているからである。しかしながら、プログラミングの際にソースコードと振舞いを明示的に分けて思考しながら構築することは少ない。これは、システム開発者は振舞いを意識していなくても、ソースコードから暗黙的にそのソースコードがどのように振る舞うかを考えているためである。

しかし通常の学習では、(1a) 学習者が明示的に機能とソースコードの有意味なかたまりを関係付けて部品知識として獲得し、(2a) その部品知識を拡張する、といった学習は行われない (図 4.3)。すなわち、ソースコードの各行でみれば何をしているかは分かるものの、そのソースコード (構造) が全体として何を達成しているのか (機能) を意識して学習しておらず、意識していたとしてもそれを関係付けて獲得しようとしていない。そのため、(1b) 学習者が機能とその機能がどのようなソースコードで実現されているかを明示的に関係付けられないことから部品知識を獲得できず、(2b) 機能を拡張したとしても内包される部品知識やその部品知識同士の関係性を意識できない (図 4.3)。

したがってプログラミングに重要な要素として、「機能」、「振舞い」、「構造 (ソースコード)」の 3 要素が位置付けられる。図 4.4 をもとに、これらの「機能」と「振舞い」、「構造」の関係性を説明する。

まず機能から構造を作り出すことを考えると、「機能」は「目標に基づいて解釈したもの」であるため、解釈後の結果でどのような「振舞い」、すなわち「入出力の状態遷移」が要求されているのかを検討する必要がある。この検討を図 4.4 では「要求」と表現している。さらに、望ましい振舞いは、どのような構造によってもたらされるものである (表現されるものである) かを検討する必要がある、この検討を図 4.4 では「表現」としている。例えば、「スワップ」の機能を達成するためには、「変数 A が変数 B の値になる」、「変数

Bが変数Aの値になる」振舞いが求められる。この振舞いを生成する構造（ソースコード）を表現すると、振舞いでは求められなかった変数Cが構造で必要となる。この一連のプロセスを経て、ある「機能」に対してどのような「構造」が対応しているかを「割り当てることができる。

一方、構造から機能を満たすことを考えると、ある「構造」に基づいて処理を行ったとき、結果としてどのような「振舞い」が生成されるかを解釈する必要がある。これを図4.4では、構造が振舞いを「生成」するとしている。さらに、そこで生成された「振舞い」を目標に基づいて解釈したときに、要求された「機能」を満たしているかを評価する必要がある。これを図4.4では「満足」とラベリングしている。この一連のプロセスを経て、ある「構造」が「機能」を「達成」しているのかを評価できる。

以上より、プログラミングの際には、達成される「機能」と機能を実現する「振舞い」、

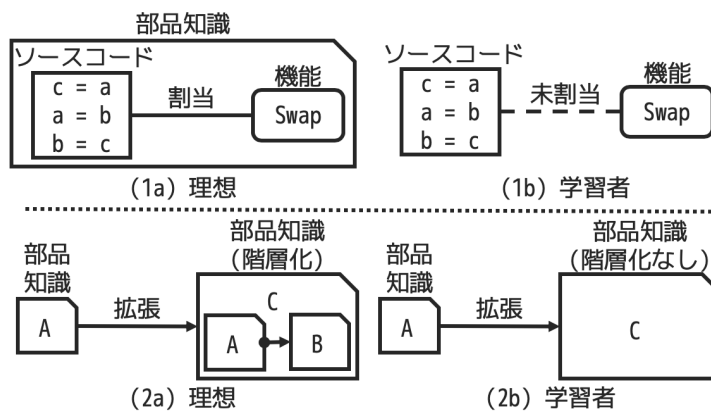


図 4.3 部品知識の獲得・拡張活動における理想状態と一般的な学習者

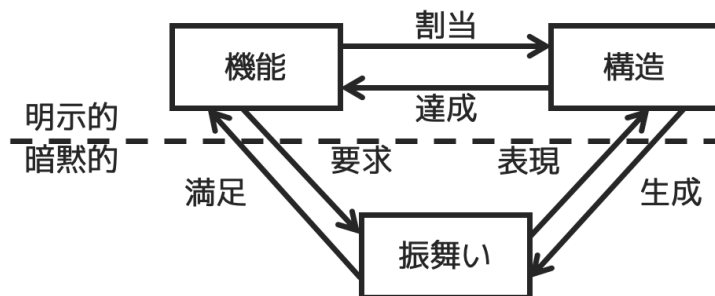


図 4.4 部品知識に含まれる要素間の関係性

振舞いを生成する「構造（ソースコード）」を関連付けて覚えていることが望ましい。よって本章では、この3要素のかたまりを部品知識として定義した。

先述の議論からも、通常、機能と構造の間に振舞いを介していることは暗黙的である。振舞いを明示的に扱わずとも、構造に機能が割当てられていれば、機能から索引して利用することが可能である。しかし、図4.2で述べた物流管理システムの例のように、「梱包管理」という偶然共通の機能（解釈後の結果）として表現されていて、偶然同じ構造によって機能が達成できる場合は、機能の再利用が可能となるが、B社ではパッケージ管理と機能を表現していたり、あるいはB社で表現する「梱包管理」がA社のものとは異なる振舞いを要求する場合、機能と構造の関連付けだけでは再利用することができない。このようなケースにおいても、知識を再利用するためには、修得済みの機能と現在必要とされている機能のそれぞれについて、振舞いと構造を関連付けて知識として獲得している必要があり、同じ振舞いを要求しているのかを判断したり、違う振舞いの場合にはどのような構造の差分が必要かを検討する必要があることになる。

また、部品同士を合成して拡張するとき、合成した部品同士はプログラムにおける処理の順序関係のもと合成される。例えば代入文を合成してスワップの機能を作るとき、 $a = b$, $c = a$, $b = c$ と部品を無造作に組み合わせるだけでは、スワップを達成できない。よってスワップを達成するには、各部品がどのような順序で実行されるかが重要となる。この処理の順序関係を、本章では部品同士の関係性と呼ぶ。よって部品同士の合成は、内包される部品とその部品同士の関係性で表現される。

本章は4.2で述べたように、プログラミング言語独立に各要素を明らかにして整理することを目的としている。一方、部品のプリミティブは対象言語におけるプリミティブな表現に依存する。そのため本章はこの目的の達成を主として、部品のプリミティブについては直接の定義はしない。したがって、以降プリミティブな部品と呼ぶときは、対象言語におけるプリミティブな表現を指すものとする。

4.4.2 部品の構築に関する考察

本節では、部品知識の獲得・拡張活動プロセスのモデル化のために、部品の構築がどのように行われるのかを考察する。

部品を作るときには、単一の構文なども含め部品を組み合わせてより大きな部品を作ることになる。この組み合わせ活動において、部品同士の関係性を構築することも必要であるのは 4.4.1 で指摘した通りである。

そのため部品の構築活動では、まず作りたい部品（以下、目標部品と呼ぶ。）の機能を一度 Decomposition（機能分解、図 4.5(a)）する思考を行い、その機能を達成するのにどのような部品（以下、構成部品と呼ぶ。）とその関係性がよいかを考えることが要求される。この活動は、いわゆるプログラミングの設計時に要求される活動に相当する。

その後、Composition（機能合成、図 4.5(b)）するとき、Decomposition によって考えた構成部品とその関係性で機能が達成できるかを実世界で構築して検証することが要求される。こちらの活動は、いわゆるプログラミングの実装時に要求される活動であるといえる。

実際に Composition してみると、思っていた機能ではなかったり（図 4.5(b), A2）、より細かい部品の組み合わせが必要だったり（図 4.5(b), C）、何らかの制約（仕様）で実現できなかったり（図 4.5(b), Target）、もしくは他の機能を実現したりということがありうる（図 4.5(b), Target'）。例えば、Decomposition の段階では A2 の部品が使えると思っていたのに、いざ実装してみると思っていた機能と違ったため、A2' を用いることで機能を代替しなければならなかった場合や、単純ソートを Decomposition し、改変しながら Composition することでバブルソートを実現できた場合などである。一方で、Composition 過程における B（図 4.5(a)）のように詳細を知っていれば（すなわち部品として獲得していれば）、Decomposition の過程においても B の詳細を組み立てる思考その

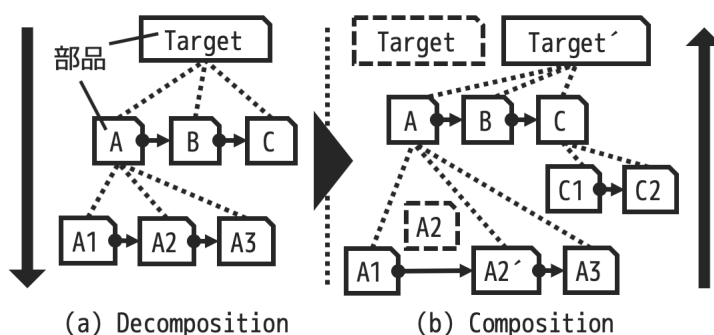


図 4.5 Decomposition と Composition の流れ

ものが省略できる。

4.4.1でも述べたように、達成する「機能」と振舞いを生成する「構造（ソースコード）」だけ覚えていれば、「振舞い」を意識するか否かは学習者に委ねられるものの、ひとまずは使うことができる。よって、部品の活用という観点からは必ずしも振舞いが必要となるわけではない。

これらの考察が示唆することは、学習者はいきなり機能から構造を作り出しているわけではなく、学習者の中で表象として設計が行われているということである。さらに、主に学習支援が行われているのは実世界の活動（Composition 過程）だけであり、学習者の表象での活動（Decomposition 過程）について支援が行われていない、という現状も示唆される。

4.4.3 部品知識の獲得・拡張活動プロセスモデル

4.4.1では部品の定義を、4.4.2では部品の構築活動をそれぞれ議論してきた。そこで本節ではこれらを踏まえて、部品知識の獲得・拡張活動のプロセスモデルを提案する。このプロセスモデルに基づいて、具体的にどのようにして部品知識が獲得・拡張されていくかを詳細に説明する。

部品知識の獲得・拡張活動プロセスモデルを図4.6に示す。なお、図中の振舞いの表現は、振舞いを条件付きで判断するために有効なUMLのステートマシン図の表記に従う。図4.6では、目標の部品知識として「2変数の並べ替え」の獲得を想定しており、学習者は既に知識として「2変数の入れ替え」の部品知識を持っていると仮定したときの、その部品知識の拡張・獲得を表現している。4.4.1で述べたように、プリミティブな部品は対象言語に依存する。そのため、この例でのプリミティブな部品は $\text{if } x > y$ を、変数 x に代入された値が変数 y に代入された値よりも大きい小さいかを判定する部品として、 $x = y$ を変数 x に変数 y に代入された値を代入する部品として説明のために用いている。よって図中では、下記の仕様で表現されている。

- 対象となるプリミティブな言語を用いたときに目標の機能達成に必要な変数群を d_n で表す

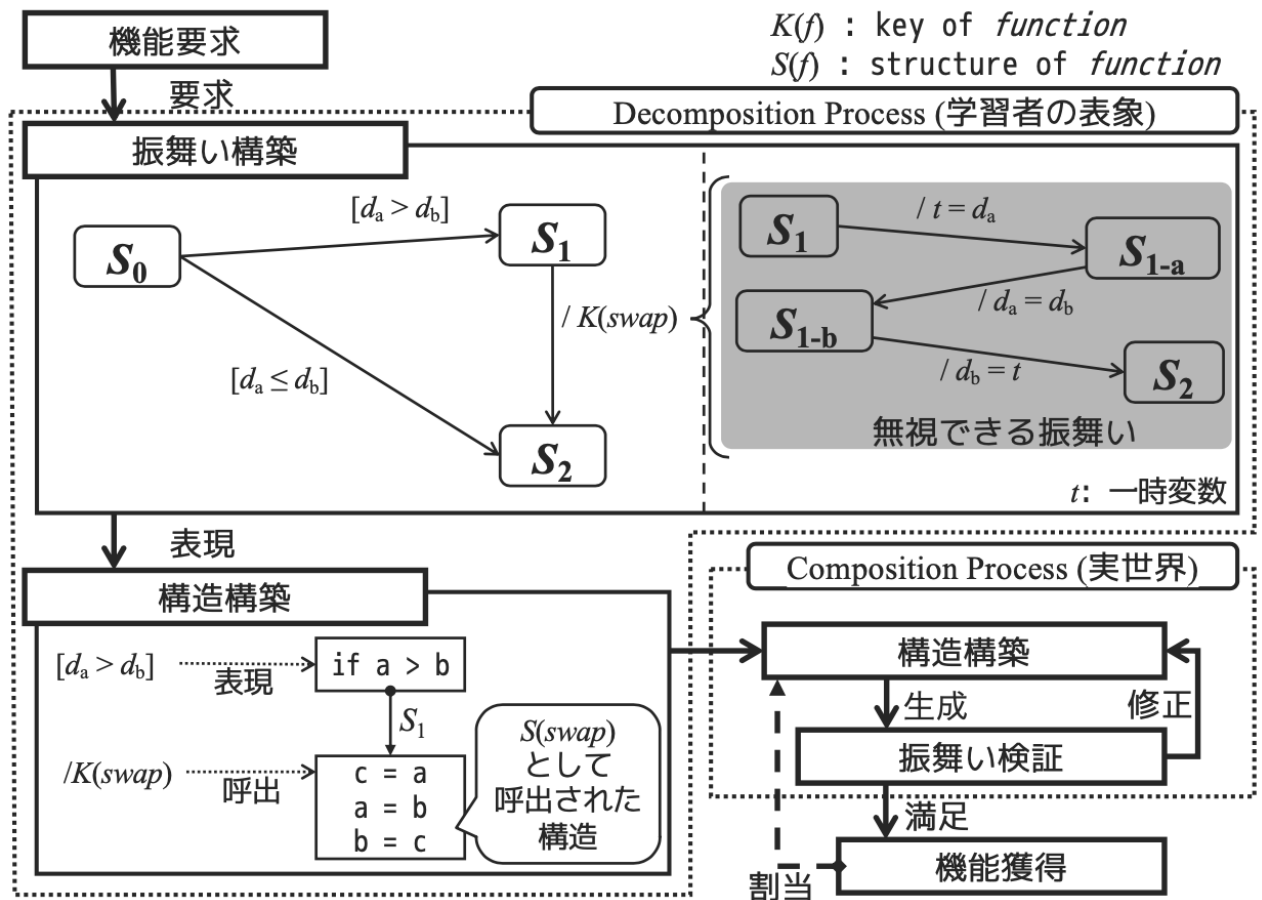


図 4.6 部品知識の獲得・拡張活動プロセスモデル
 (「2 変数の入れ替え」を拡張して「2 変数の並べ替え」を獲得する場合を事例に)

- 変数は図中では値型のみだが、文字列型などプリミティブな言語に依存する形で型が表現される
- ある状態 S_n は、その時点における変数群 d_n の値を状態とした集合とその変数が満たす条件 ($[a > b]$ など) で構成される
- 初期状態 S_0 は、予め学習者に対して問題文などで与えられた入力か、学習者が振舞いからその機能の達成を検証するのに必要だと推測した入力が行われている状態であり、この状態では変数がどのような条件を満たしているか判別していない状態である
- 状態 S_n と S_m を繋ぐエッジ上では、ステートマシン図の記法に従い「イベント [条

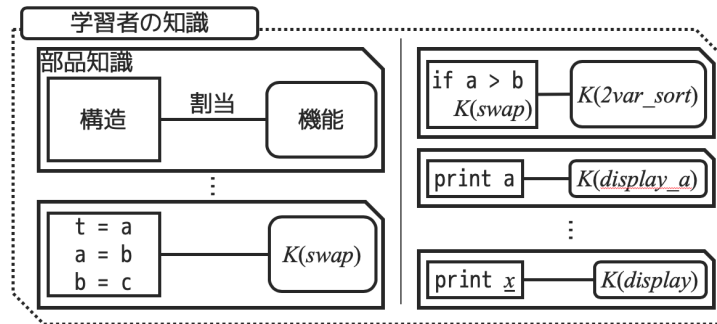


図 4.7 学習者の知識状態の表現

件]/アクティビティ」の表記で表す（ただしいずれも省略可能）

- 本章では遷移条件を大小比較しか取り扱っていないものの、一般的な if 文と同等の扱いであり、`==`などのオペランドもプリミティブな言語に依存する形で取り扱われる
- $K(\text{function})$ はある機能 `function` の構造を呼び出す索引の意味であり、学習者の既有知識を用いることによる表象する操作過程の省略に用いられる
- $S(\text{function})$ はある機能 `function` の索引をもとに呼び出される構造であり、 $K(\text{function})$ と対応している

図 4.6 では、プロセスは大きく二つに分かれる。一つ目のプロセスは、学習者内の表象で行われる Decomposition プロセスである。二つ目のプロセスは、実世界で行われる Composition プロセスである。

Decomposition プロセスでは、まず、目標の機能を達成するためにどのような振舞いが必要かを考え、表象として構築する。この際、入力された変数 (d) の集合が初期状態 S_0 として表現されるが、明示的に与えられた入力がない場合、学習者が振舞いからその機能を検証するのに必要だと推測した入力を用いることになる。この際、仕様として変数の制約が与えられない限りは（たとえば、 $a > b$ という前提などがない場合は）、データセットに制約はないものとなる。 S_0 から他の状態に移行するときは、ステートマシン図の記法にならい、エッジ上の矩形内にイベント [条件]/アクティビティの形で移行時に発生したイベントやそのイベントの条件、そのイベントが発生したときのアクティビティを記述する。イベント、条件、アクティビティはいずれも省略可能であり、ある条件を満たすとき

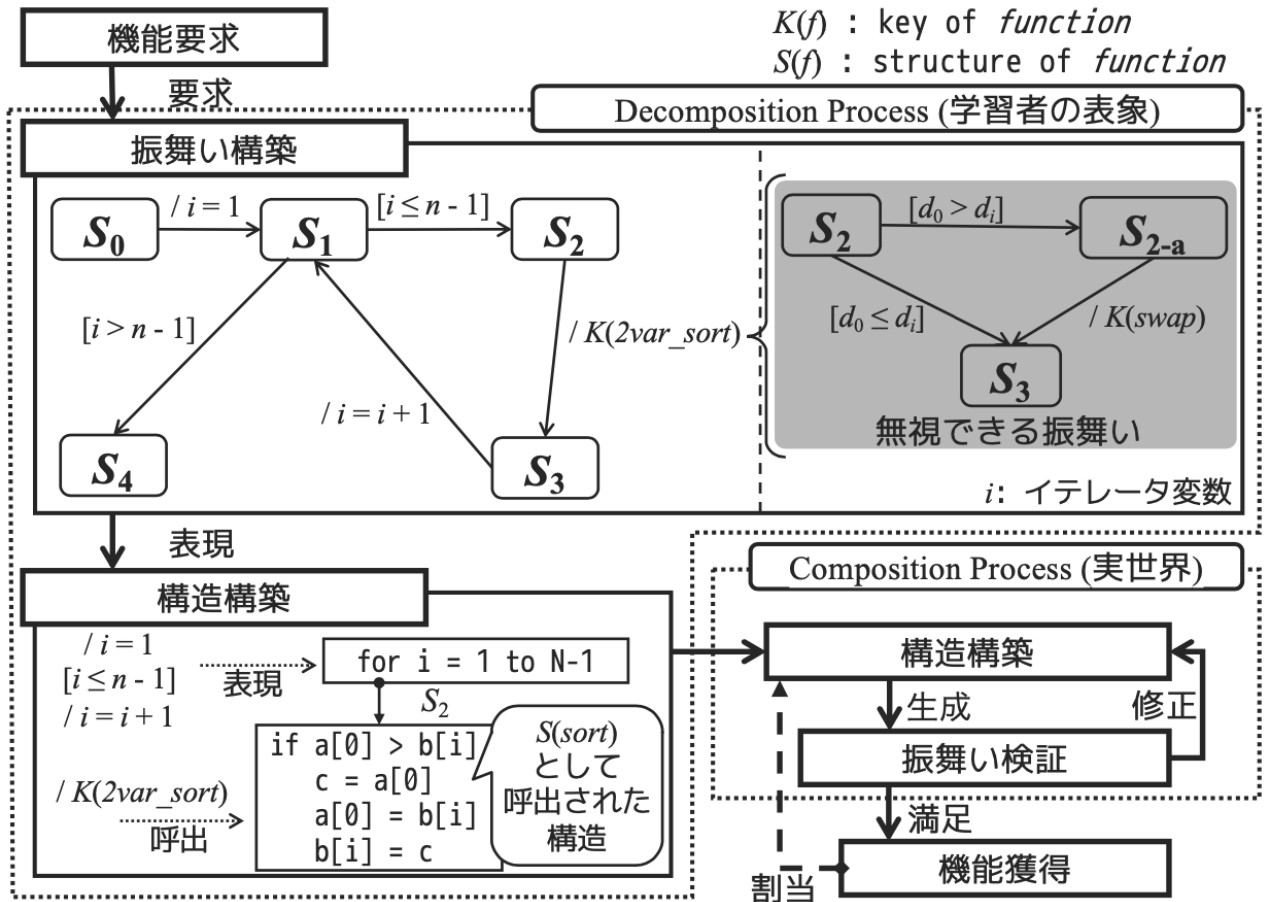


図 4.8 部品知識の獲得・拡張活動プロセスモデル
 (「2 変数の並べ替え」を拡張して「配列の最小値を求める」を獲得する場合を事例に)

のみの遷移や、条件によらない遷移、およびその際のアクティビティだけを記述するなど多くの用途に耐えうる記法となっている。そのため、図 4.6 内の振舞いについて解説すると、 S_0 は処理前の状態であり、変数 d_a と d_b の制約がない状態（大小関係が未定の状態）である。この状態より、 $d_a \leq d_b$ という条件を満たした場合、状態 S_2 ($d_a \leq d_b$ の制約が成立した状態) に移行する。また、 S_0 より、 $d_a > d_b$ を満たした場合、状態 S_1 ($d_a > d_b$ の制約が成立した状態) に移行する。この状態では、 $K(\text{swap})$ の操作（アクティビティ）が行われる。ここでの $K(\text{swap})$ は「2 変数の入れ替え」の構造を呼び出す索引の意味であり、学習者の既有知識を用いることによる表象する操作過程の省略に用いられる。知識を持たない学習者であれば、図 4.6 の右上灰色部に表されるような、詳細な操作を想定し

ないと達成できないが、「2変数の入れ替え (swap)」の構造をあらかじめ知っていることで詳細な振舞いについて考慮する必要がなくなる。

必要となる振舞いが決まると、次に振舞いを構造として表現することで構成部品を組み立てる。このとき、振舞いを直接プリミティブな部品で表現するか、既有知識から部品知識を索引する場合は構造を呼び出す。まず、「 $d_a > d_b$ 」はif文としてそのまま表現することができる。次に、「 $K(\text{swap})$ 」では swap の索引をもとに学習者の既有知識から構造を呼び出すことで、swap の構造が作られる。このとき、if 文の構造と swap の構造の間には、 S_1 の状態が介在しているといえる。

次に、Composition プロセスについて説明する。学習者の表象で構造ができると、その表象での構造をもとに実世界に構造をソースコードとして書き出す。そして、ソースコードを実行することにより振舞いを生成し、検証する。このとき、求める振舞いでなければ実世界の構造を修正することになる。最終的な構造の振舞いが機能を満たすことができれば、学習者はその構造に機能を「割当」てることで知識として獲得することになる。そのため、学習者の知識状態は部品知識の集合として表現できる (図 4.7)。例えば図 4.6 上の Decomposition プロセスにおける構造の構築では、学習者の既有知識から $K(\text{swap})$ の機能が「割当」られた部品知識を索引し、構造を呼び出している。さらに、図 4.6 によって追加された「2変数の並べ替え」は $K(2\text{var_sort})$ として新しく知識に加わる。図 4.6 のプロセスから「2変数の並べ替え」を獲得した学習者は、その後、図 4.8 に示すプロセスで「配列の最小値を求める」の獲得を目指す。このようにして、部品を段階的に拡張することで、部品知識の体系化を行っていく。

ここでの索引に用いる機能名 $K(\text{name})$ は、学習者の中で部品知識の区別に用いられるラベルに過ぎないため、特別な意図をもたない。

学習者の知識は、部品知識の獲得・拡張プロセスを通じて特定の構造に機能が「割当」てられた状態で獲得される。例えば図 4.7 の変数 a を表示する部品知識 `display_a` は `print a` の構造と結びついており、変数 b など他の変数の表示には直接用いることができない。しかし変数 b を表示する部品知識 `display_b` などを獲得することで、図 4.7 中の部品知識 `display` のような「特定しないある変数 (図中の斜体下線)」を表示する部品知識に一般化できるのではないかと著者は考えている。

4.3 で述べたように、古池らによる研究 [Koike 18b, Koike 18c] では、このような部品知識の一般化—特殊化や全体化—部分化の関係性に着目したグラフ構築に関する議論が行われている。しかし古池らの研究では、4.4.1 で述べたような部品知識内における機能—振舞い—構造の区別、特に機能と振舞いの差は十分に議論されていない。また、グラフの接続関係から次に学習すべき部品知識を提供することができるものの、構造—振舞い—機能を結びつけるための学習支援方法までは述べていない。よって本章における機能—振舞い—構造の区別に関する議論や、それらを結びつける学習支援方法に基づき、今後こういった関係性を考慮したグラフ構築を行うことが漸進的な学習に効果的であると考えられる。

4.5 提案モデルの利用に関する考察

本章で提案する部品知識の獲得・拡張活動プロセスのモデル（以下、提案モデル）により、どのように学習活動に貢献できるのかを検討することは、提案モデルの有用性を示すうえで重要だと考えられる。

よって本章では提案モデルに基づくことにより、どのような知的学習支援システム (ITS: Intelligent Tutoring System) が実現可能になるのか、実現した ITS がどのように学習活動に対して貢献が可能なかを考察する。

提案モデルの特色をまとめると、(i) 部品知識を取り扱えることと、(ii) 部品知識の獲得・拡張過程を二つのプロセスごとに3要素の側面から捉えられることにより、知識の獲得・拡張過程が捉えられることにある。したがって提案モデルに基づくことにより、ITS は以下の機能を持つことができる。(1) 部品知識の管理機能、(2) 構造（ソースコード）の振舞い捕捉機能、(3) 振舞いに基づく機能の同定機能、(4) 誤り箇所の同定機能、の4点である。以下では、それぞれの機能とそれによって実現される学習活動への支援について考察する。

4.5.1 部品の管理機能

提案モデルに基づくことにより、ITS が知識を部品知識として管理することができるようになる。すなわち、ITS は図 4.6 のような機能—振舞い—構造を部品知識として把握し

ていることになる。これにより、学習者に対してある機能の振舞いや構造を解答させる問題（構築課題）や、ある構造の振舞いや機能を解答させる問題（読解課題）を出題することが可能となる。

さらに、ITS が知識として部品知識の集合を持っていることで、ある部品知識に内包される部品知識同士の関係性が把握できているため、半順序構造として部品知識全体の関係性を把握している。よって、学習者が解けた問題を獲得した部品知識とみなし、学習者がどの部品知識を持っているかを把握することで知識状態を推定することができる。また、この知識状態をもとに問題を提示して問題に解答できなかった場合は、誤った箇所の部品知識にバグがあるとみなし、その部品知識に対する理解のための補助問題を提示することが可能になる。

4.5.2 構造（ソースコード）の振舞い捕捉機能

提案モデルに基づくことで、学習者の構造（ソースコード）の振舞いを捉えられるようになる。従来からデバッガなどにより、各行の操作ごとに対象の状態を捉えることは可能とされていたが、デバッガ自体は知識を持たないため、構文規則が合っていれば誤りを指摘することはなく、どのような構造が望ましいかを診断することはできない。

一方、提案モデルではデバッガなどから対象の状態と前後の操作を捉えることで、振舞いレベルで学習者の解答を捉えることができる。これと 4.5.1 により、学習者の解答した構造に依存しない形で学習者の誤り箇所を同定することが可能となる。

これらを、構築課題を想定して事例 1 に示す。

事例 1

提示問題：2 変数の入れ替えの機能を作りなさい

学習者の解答した構造：

$x = y$

$y = x$

正誤判定：誤り

誤り箇所フィードバック：

「1 行目では一時変数 x に入力変数 y が代入されていますが、2 行目では、 y にもう一方の入力変数を代入する必要があります。」

この事例 1 では、正解の構造しか持たないシステムにおいて正解が「 $c = a$ 」, 「 $a = b$ 」, 「 $b = c$ 」である場合に、1 行目から誤りになってしまう。しかし提案モデルに基づくことにより、「 $/t = d_a$ 」, 「 $/d_a = d_b$ 」, 「 $/d_b = t$ 」として振舞いを捉えることができ、捕捉した振舞いから学習者の解答した変数名に依存することなく構造の 2 行目に誤りがあると推定することができる。

4.5.3 誤り箇所の同定機能

提案モデルでは、Decomposition プロセスと Composition プロセスに分けて学習者の活動が把握できる。したがってこれと、4.5.1 と 4.5.2 とを組み合わせることで、部品の獲得・拡張プロセスの間のどこで誤ったかを同定する機能が実現できる。これにより、学習者に振舞いと構造を個別に解答させることで、学習者が Decomposition のプロセスにおいて機能から要求される振舞いへの変換が誤っているか、振舞いから表現する構造への変換が誤っているかを判断し、それに対してどこがどのように誤っているのかを学習者に説明することができる。これを以下の事例 2 に示す。

事例 2

提示問題：2 変数の並べ替えの機能を作りなさい

学習者の解答した振舞い：

$S_0 \rightarrow (/t = d_a) \rightarrow S_1 \rightarrow (/d_a = d_b) \rightarrow S_2 \rightarrow (/d_b = t) \rightarrow S_3$

学習者の解答した構造：

t = a

a = b

b = t

部品知識に基づく正しい振舞い：

(図 4.6 中の「振舞い構築」参照)

誤り箇所判定：Decomposition 中の機能→振舞い

誤り箇所フィードバック：

「Decomposition プロセスにおける機能から要求される振舞いへの変換が間違っています。2 変数の並べ替えでは、 $a \leq b$ が成り立つときには何も実行する必要がありません。そのため、 $a \leq b$ なのか $a > b$ なのかを 1 行目に判定する必要があります。」

ただし上記の機能により Decomposition プロセスの支援を過剰に行ってしまうと、Decomposition で導き出された正しい構造を Composition で生成するだけの活動になってしまい Composition のプロセスにおいて学習者が試行錯誤を通した修正活動を十分に経験できず、構造の振舞いを観察して自身の構造を振り返るといった、試行錯誤のプロセスとして期待した活動が行われないといった問題点がある。これは、構造に基づく振舞いの生成と観察が実世界で行われる Composition でしか経験できないことに由来する。そのため、ITS は Decomposition プロセスと Composition プロセスが存在することを把握しながらも、Decomposition での支援の量を調整することで、学習者の試行錯誤を促す Composition のプロセスを行わせる必要があるといえる。

4.5.4 振舞いに基づく機能の同定機能

学習者の解答した構造の振舞いを捕捉できるのは 4.5.2 で説明した通りだが、さらに捕捉した振舞いを 4.5.1 における部品知識の集合と同定することで、どの機能が達成されて

いるのかを同定することができる。

これらを、構築課題を想定して事例 3 に示す。

事例 3

学習者の解答した構造：

```
if x > y
  a = x
  x = y
  y = a
  a = b
```

機能同定フィードバック：

「この構造では、“2 変数の並べ替え”の機能が達成されています。」

事例 3 における「a = b」は、「2 変数の並べ替え」の機能を最小限の構造で実現する場合には冗長な構造だと考えられる。しかし、「2 変数の並べ替え」の機能が必要とする振舞いを満たしていることから、このような冗長な構造に対しても十分条件として柔軟に診断することが可能となる。

しかしこの機能が適切に学習者の解答した構造や振舞いから機能を同定するためには、前提として ITS が対象範囲においてある程度包括的な部品の集合を知識として持っている必要がある。例えば、ソートの範囲が対象であれば ITS が「配列の並べ替え」の機能の振舞いが異なる部品知識を複数持っていないと、「バブルソート」に対応する振舞いや、「マージソート」に対応する振舞いに対して、それが「配列の並べ替え」の機能であると同定することができなくなってしまうためである。

4.6 説明機能に関する評価実験

4.3 で述べたように、古池らは、部品知識の獲得を促す学習手法を備えた学習支援システムを開発し、その学習効果を確認している [古池 18a]。さらに、システムのフィードバックが学習範囲に対して有効に働くことも検証している [古池 19a]。しかし、古池らのシステムでは 4.5.2 のような振舞いの捕捉はできておらず、システムは構造-機能の対応関係のみで部品知識を取り扱っていたため、学習者の構造と正解の構造とを比較することで不足部分の指摘を行っていたに過ぎない。一方、提案モデルに基づき部品知識に対して

振舞いの要素を加えることで、古池らのシステムではできなかったフィードバックが可能になると考えている。

したがって、提案モデルによって実現できる ITS の機能のうち、4.5.1 で説明した部品知識の管理機能と、4.5.2 で説明した構造の振舞い捕捉機能に基づくことで実現できるフィードバックの有効性を検証するため、古池らのシステム [古池 19a] で行っていたフィードバックと比較する評価実験を行った。

4.6.1 実験方法

実験参加者はプログラミングの講義で最低3年間プログラミングを学んだ工学系大学生と大学院生の計17名である。また実験参加者らは、for文、if文などの基本的な概念や、ソートなどのアルゴリズム、関数の学習については修了しているものである。

実験方法は、まず実験参加者に想定される学習者の誤りとそれに対する説明を2種類与え(表4.1)、どちらの説明が良いかを表4.2に示す質問ごとに、4種類の説明に対して評価してもらった。これらの説明は、一つは古池らの学習支援システム [古池 19a] で提示されていた説明と、もう一つは提案モデルに基づくことで生成できる説明である。例えば、表4.1に示す事例の提案モデルに基づく説明ではシステムが部品を捉えられているために、学習者の解答から推測される部品と同定して説明している。

評価の方法としては、[古池 19a] のシステムで提示される説明を1側、提案モデルに基づく説明を5側の両端に置いて1~5の数字を示す形で5件法を用いて、質問ごとに適していると感じた説明を選択することで評価してもらい、その理由についても記述してもらった。

4.6.2 実験結果と考察

実験結果について、5件法において提案モデルに基づく説明にポジティブな評価(4, 5)をする場合と、そうでない評価(1, 2, 3)をする場合が同じ確率で起こることを帰無仮説とした。そのため以後本文中では、提案モデルに基づく説明にポジティブな評価(4, 5)を Positive, そうでない評価(1, 2, 3)を Not Positive として示し、その数を比較して議

表 4.1 比較する説明の例（説明 3）

説明 3	
問題	「2 変数を並べ替えて出力する機能を作りなさい」
初心者の回答例	<pre> if a > b a = b b = a print a print b </pre>
[古池 19a] のシステムで提示される説明	2 行目は、「c = a」が正解です.
モデルの説明	この機能は、「2 変数の並べ替え」と「2 変数の出力」の組み合わせでできています. そのため「2 変数の出力」は合っていますが、「2 変数の並べ替え」では、2 行目で a に b の値を代入する前に a の値を一時変数に代入する必要があります.

表 4.2 評価実験で用いた質問項目の一覧

質問項目
(1) あなたが、プログラミング初心者がこのような間違いをしたときに説明するとしたら、どちらが今後のプログラムの深い理解に影響を与えと思いますか？
(2) あなたが、プログラミング初心者がこのような間違いをしたときに説明するとしたら、どちらが今後プログラムを自分で作れるようになることに影響すると思いますか？
(3) あなたが、プログラミング初心者がこのような間違いをしたときに説明するとしたら、どちらが今後なぜ間違っただかを自分で気づけるようになることに影響すると思いますか？
(4) あなたが、プログラミング初心者がこのような間違いをしたときに説明するとしたら、どちらが初心者に対する説明として適切だと思いますか？
(5) あなたが、プログラミング初心者がこのような間違いをしたときに説明するとしたら、どちらが説明としてわかりやすいと思いますか？

論する. また、分布を示すために最頻値と中央値を用いる.

合計 4 つの説明に対する評価を合算した概要について表 4.3 に示す. 表 4.3 より、説明 1~4 の全ての質問において、提案モデルに基づく説明が高い評価を得た.

次に、各説明とその評価の結果、また片側二項検定の結果を表 4.4 に示す. 片側二項検定では、評価項目ごとに Positive と Not Positive の数に対して検定を行った. 検定の結

表 4.3 実験結果の概要 (1: 従来 [古池 19a] の～5: 提案モデルに基づく説明)

$N = 68$	(1) 理解	(2) 開発	(3) 探索	(4) 適切	(5) 平易
Positive	55	56	58	48	50
Not Positive	13	12	10	20	18
中央値	5	5	5	5	5
最頻値	5	5	5	5	5

果を表 4.4 中にて $^+(p < .10)$, $^*(p < .05)$, $^{**}(p < .01)$ にて示す. 検定の結果からも, 説明ごとの結果においても全ての質問において概ね提案モデルに基づく説明が有意に高い評価を受けていることが分かる.

表 4.4 各事例ごとの評価 (1: 従来 [古池 19a] の～5: 提案モデルに基づく説明)

$N = 17$	(1) 理解	(2) 開発	(3) 探索	(4) 適切	(5) 平易
説明 1					
Positive	14	16	14	12	13
Not Positive	3	1	3	5	4
有意差	**	**	**	+	*
中央値	5	5	5	5	5
最頻値	5	5	5	5	5
説明 2					
Positive	15	14	16	11	11
Not Positive	2	3	1	6	6
有意差	**	**	**		
中央値	5	5	5	5	5
最頻値	5	5	5	5	5
説明 3					
Positive	14	12	13	12	13
Not Positive	3	5	4	5	4
有意差	**	+	*	+	*
中央値	5	5	5	5	5
最頻値	5	5	5	5	5
説明 4					
Positive	12	14	15	13	13
Not Positive	5	3	2	4	4
有意差	+	**	**	*	*
中央値	5	5	5	5	5
最頻値	5	5	5	5	5

しかしながら, アンケートの評価のみでは (i) 真面目に答えていない, (ii) 真面目に答えようとしているが特徴を捉えて評価できていない, といった懸念が残る. したがって, (a) アンケートの各評価項目への回答が適当である, (b) 回答の理由記述が特徴を捉えている, といったことを示し, アンケートの回答に対する信頼性を確かめる.

まず, 各説明に対する評価の信頼性を測るため, 各質問項目への評価についてクローン

バックの α 係数を算出した。その結果、 α 係数が 0.96 となり、高い信頼性を示した。したがって、質問項目には一貫性があるとみなせる。

また、それぞれの評価に対する理由記述の量が多い人ほど真面目に答えている可能性が高く、回答の信頼性が高いと仮定することができる。よって理由記述量の上位群 8 名と、下位群 9 名に分けて群間の評価に差があるかを調査した（表 4.5）。表 4.5 より、Positive と Not Positive の数がほぼ同等の割合であり、中央値と最頻値からも分布に差がないと考えられる。このことから、文章量によって評価に差はみられなかった。

表 4.5 記述量の上位群と下位群における評価

	上位群の評価 (N = 8)	下位群の評価 (N = 9)
Positive	120	147
Not Positive	68	59
中央値	5	5
最頻値	5	5

さらに、評価対象ごとに特徴を捉えて評価することができるかを検証するため、回答に対する理由記述文章に対してクラスタ分析を行った。クラスタ分析では、[古池 19a] のシステムで提示される説明群と提案モデルに基づく説明群はそれぞれ独立した評価基準で評価されていると仮定すれば、それぞれの説明群のみのクラスタ分析では異なる特徴を表すクラスタが形成されることが期待できる。さらに、全評価理由でのクラスタ分析でそれぞれの説明群のポジティブな特徴を捉えたクラスタが確認できれば、全体においてもそれぞれの説明群が混濁して評価されたのではなく、独立した基準で評価されたと仮定できる。したがって、[古池 19a] のシステムで提示される説明群と提案モデルに基づく説明群、全評価理由の三つの群でクラスタ分析を行う。クラスタ分析のために、まず、全評価項目に対する理由 340 件から、[古池 19a] のシステムで提示される説明にのみ言及した評価理由 67 件（[古池 19a] のシステムで提示される説明群）と提案モデルに基づく説明にのみ言及した評価理由 220 件（提案モデルに基づく説明群）を評価の点数を問わず抽出し、それぞれの文章群に対して、形態素解析を行った。形態素解析では、名詞、形容詞、形容動詞、副詞、動詞を抽出した。その結果、全体で 297 語、[古池 19a] のシステムで提示される説明のみに言及した評価理由から 88 語、提案モデルに基づく説明のみに言及した評価理由から 204 語が抽出された。それぞれのクラスタ分析では、さらにこれらの抽出され

た語から出現数が n 回以上の単語をベクトルとしてそれぞれのベクトル間で Jaccard 係数を求めた。この Jaccard 係数を距離として、Ward 法によるクラスタリングを行った。[古池 19a] のシステムで提示される説明のみに言及した評価理由では出現数 2 回以上の語である計 38 語をベクトルとし、提案モデルに基づく説明のみに言及した評価理由では出現数 5 回以上の語である計 50 語をベクトルとした。また、全体の評価理由では出現数 5 回以上の語である計 75 語をベクトルとした。クラスタ分析では、それぞれの併合水準に基づきクラスタ数を決定し、[古池 19a] のシステムで提示される説明群を 7 クラスタ、提案モデルに基づく説明群を 9 クラスタに決定した。また、理由記述文章全体におけるクラスタ分析を、併合水準をもとに 12 クラスタとなるように行った。

それぞれのクラスタ分析の結果を表 4.6 に示す。表 4.6(a) より、[古池 19a] のシステムで提示される説明に対しては第 1 クラスタに「単純」、「短い」、「明確」などがみられることや、第 4 クラスタに「間違える」、「気付く」、「自分」などがみられる。文章が短く、かつ単純であり説明が少なく、自分自身で間違いに気付く可能性がある、という [古池 19a] のシステムで提示される説明の特徴を捉えたまとまりがみられる。また表 4.6(b) より、提案モデルに基づく説明に対しては第 3 クラスタに「丁寧」、「理解」がみられることや、第 9 クラスタに「分かる」、「詳細」、「機能」、「構造」などがみられる。説明が丁寧であり理解の促進に繋がることや、機能と構造の関係の詳細が分かる、という提案モデルに基づく説明の特徴を捉えたまとまりが確認できる。さらに実際の自由記述文の代表例として、[古池 19a] のシステムで提示される説明に対して Positive / Negative な文章、提案モデルに基づく説明に対して Positive / Negative な文章をそれぞれ調査した。その結果、[古池 19a] のシステムで提示される説明に対して Positive な文章の代表例としては「答えを教えてくれたほうが分かりやすいから」という回答があり、一方、[古池 19a] のシステムで提示される説明に対して Negative な文章の代表例としては「答えだけを提示したのでは次に繋げることが出来ない」という回答があった。また、提案モデルに基づく説明に対して Positive な文章の代表例としては「プログラムの仕組みを理解したほうが、今後間違いが分かるようになるから」という回答があり、提案モデルに基づく説明に対して Negative な文章の代表例としては「提案モデルに基づく説明は初心者にとっては難しいと思う」という回答があった。したがって、[古池 19a] のシステムで提示される説明群と

提案モデルに基づく説明群はそれぞれ独立した評価基準で評価されていると考察できる。

さらに、理由記述文章全体におけるクラスタ分析の結果を表 4.7 に示す。表 4.7 から、第 8 クラスタに「考える」、「必要」、「学習者」、「直接的」とみられることや、第 10 クラスタに「分かる」、「シンプル」とみられることから、[古池 19a] のシステムで提示される説明の特徴を捉えたクラスタが確認できる。また一方で、第 2 クラスタに「説明」、「機能」、「細かい」とみられることや、第 5 クラスタに「丁寧」、「説明」、「理解」とみられることから、提案モデルに基づく説明の特徴を捉えたクラスタも確認することができる。よってこれらから、理由記述文章全体においても、[古池 19a] のシステムで提示される説明の特徴を捉えたクラスタや提案モデルに基づく説明の特徴を捉えたクラスタが確認できた。したがって、それぞれの特徴を捉えた理由記述は、一定の考えに基づく記述と捉えられる。

これらの結果から、古池らが提案した部品知識を学習する手法を用いたシステムにおいて、提案モデルに基づくことで従来よりも適切なフィードバックが実現できると考えられる。したがって提案モデルを用いることで、部品の学習において効果的な支援を実現できる可能性が示唆された。

一方で本実験では、評価者は本システムの背景を知らない状態で回答していることが懸念点としてあげられる。したがって、説明文の豊富さからシステムの有効性が評価されている可能性は否定できない。その一方で、そもそも従来のシステムでは豊富な説明文を生成するための情報を持っておらず、本モデルの説明のような情報量を持った説明の生成自体が困難と考えられる。その点において、豊富な説明文を生成できる本モデルは一定の有用性があると考えている。また、クラスタ分析や自由記述文の代表例から [古池 19a] のシステムで提示される説明群と提案モデルに基づく説明群には意味的な差が示唆されており、単に説明文の豊富さだけで評価されたのではなく、説明内容についても考慮されたものと考えられる。この点について明らかにするために、今後は説明文の文章量を統制した状態での追加調査が望まれる。

4.7 議論

本章では、4.7.1 にて本章の位置付けについて、4.7.2 にて今後の展望について議論する。

4.7.1 本章の位置付け

本章の位置付けを明確にするため、いくつかの先行研究を紹介し、本章との関連性について述べる。

いくつかの先行研究では 2 章で述べたような、プログラミングにおいて chunk の存在やその重要性を主張している。

Schankin らはコードを読んで理解するうえで、命名規則が及ぼす影響について調査を行っている [Schankin 18]。調査では、短い名前よりも、長くても一定の命名規則に従った構造が分かる名前のほうが理解の助けになる、といった仮説を検証した。その結果、経験の豊富なシステム開発者に対しては有効であったものの、経験の浅いシステム開発者にとっては効果がなかった。このことについて、Schankin らは経験豊富なシステム開発者が長い名前を簡単に呼び出せたことは、長期記憶での chunking（有意味な知識のかたまりの構造化）が影響していると考察している。つまり、経験の浅いシステム開発者はそもそも有意味な知識のかたまりが形成されていないため、名前から呼び出せる有意味な知識のかたまりが少ないためだと考えられる。また Adelson は、初学者や熟達者のシステム開発者がプログラミングの概念をどのように表現し使用するかという問題について調査した結果、初学者はステートメントベースの編成を使用したがるが、熟達者はプログラム機能の原則に基づくより抽象的な階層的編成を使用したことを報告した [Adelson 81]。Weiser は、通常、システム開発者は大きなプログラムを小さなかたまりのある部分に分割して理解していると提唱し、実験を通してシステム開発者は日常的に首尾一貫した分割を行うことを検証した [Weiser 82]。ただし、なじみのないプログラムをデバッグするときはスライスと呼ばれる、データの流れに関連した一連のステートメントで分割することも実験から示した。Pennington も、プログラミングの理解において最初は手続き的な理解から始まり、徐々に機能的な理解に移行していることを調査から示した [Pennington 87]。

また、ソフトウェア工学の分野ではプログラムの理解プロセスが議論されている。Exton はプログラミングにおける問題解決モデルについて、既知の目標を分解するトップダウンの思考と、未知の目標を構築するボトムアップの思考に分けられると主張している [Exton 02]。また O'Brien らは、ソフトウェアの理解プロセスは本質的にトップダウンとボトムアップとして概念化されているという前提において、トップダウンの理解が二

つに分かれることを提唱した [O'Brien 04]. 一つは予想ベースの理解であり、そのプログラムが何をしようとしているかの予想をもとにその予想と同定する形で行われる理解である。もう一つは推論ベースの理解であり、プログラムはコード内のそれぞれの部分の意味から全体において何をしようとしているかを導き出す理解である。このことについて Obaidellah の結果は、初学者クラスである CS2 において推論ベースの理解よりも予想ベースの理解のほうが容易であったことを示している [Obaidellah 16]. したがって、個別の部品知識を合成して対象を思考する推論ベースの理解ではなく、既に合成済みの部品知識を同定するほうがシステム開発者にとって容易であったと考えられる。

これらの研究が示唆していることは、最初はなじみのないプログラムの理解においてステートメントレベルから手続き的に理解しているものの、徐々に熟達していくにつれ有意義な知識のかたまりを形成し、機能的な理解に変化していくということである。さらにこの機能的な理解は、対象の理解に良い影響をもたらすことも示唆されている。認知科学の観点からもこの考え方は支持されており、Peitek らは、ソースコード理解の過程について fMRI を通じて観察した結果、ボトムアップの理解よりもトップダウンの理解のほうがプログラムの理解に割かれる認知エフォートが減少することを検証した [Peitek 20]. またプログラミング手法自体においても、Biggerstaff らによって部品化再利用という方法が提案されている [Biggerstaff 87]. これはソースコードの再利用を前提としたとき、あらかじめ部品を作成してそれらの部品を検索、変更、合成することで目的のプログラムを作成する方法である。

総じて、有意義な知識のかたまりが重要であることが主張され、その効果も報告されている。しかしながら、知的学習支援において有意義な知識のかたまりを獲得するための支援は行われてこなかった。よって本章で提案するモデルは、この有意義な知識のかたまりの獲得支援を ITS で実現するための重要な基盤となるモデルとして位置付けられる。

4.7.2 今後の展望

本節では、今後の展望について述べる。

4.3 では、本章で扱う知識の範囲を明らかにするためにプログラミング学習における知識を「部品知識」と「メタ部品知識」の二つに大別した。しかし、後者のメタ部品知識は

問題解決過程を通して獲得されることを期待しているものの、本章の範囲においては明示的に取り扱えていない。繰り返しになるが、メタ部品知識とは、部品の「構築方法」や、「獲得方法」、「拡張方法」などの部品の取扱いに関する知識である。したがって、「構築方法の知識」を身に付けたということを定性的に表現することは困難であるといえる。しかし、これらのメタ部品知識の獲得は、未知の部品を構築し、その部品知識を獲得・拡張していくうえで重要であり、何らかの形で獲得を支援できることが望ましい。著者は、これらの獲得をより明示的に支援するために、メタ認知を活性化する支援が有効であると考えている。具体的には、「自分自身が何を考えて構築しているか」、「構築を誤ってしまったとき、何が誤っているのか」、「なぜ誤ったのか」を学習者に思考させることができれば、学習者自身の構築方法や獲得方法などのメタ部品知識について振り返り、誤りを修正する機会を与えられる可能性がある。このように、メタ部品知識の獲得支援は本章の展望の一つに挙げられる。

また、提案モデルを通して知識体系化や chunk の獲得を支援するにあたり、部品知識の獲得に必要な知識と、学習者が備えておくべきプログラミングに関する知識の関係性については明らかになっておらず、今後解明が必要となっている。

部品知識の記述で表現できることについても、課題が残っている。例えば、いくつかの入出力をもつ部品は注目する入出力によって機能が変わることがある。このことについて笹島らは、「熱交換器」の振舞いにおいて「水を冷やす機能」と「水を温める機能」が存在することを例に挙げて指摘している [笹島 96]。このように部品の振舞いから機能が一意に定まらない問題について、笹島らは FT (Functional Topping) という枠組みの中で、部品の注目する振舞いごとに機能を分けて記述することで解決している。よって本章でも、FT の枠組みを参考にこの課題の解決をねらっている。

平嶋らは、従来の ITS における問題解決モデルの多くが (1) 問題解決の失敗を主として知識の誤りや不足により発生するものと表現している、(2) 問題解決能力の向上を、新しい知識の獲得として表現しているという、二つの特徴を持っていることを指摘している [平嶋 92]。また、算数の文章題など問題解決が複雑な領域については (1) (2) だけでなく、(A) 必要な知識を持っているにもかかわらず発生する問題の行詰り、(B) 既に獲得している解法についての習熟、に対する支援が必要だと主張している。本章でも同様に、

(A), (B) の支援が重要だと認識しており, 検討を重ねている.

4.8 本章のまとめ

本章では, プログラミングにおける再利用を指向した知識の体系化支援を行うための問題解決過程モデルを提案した. まず, プログラミングにおいて部品知識が何であるかを, プログラミングの要素を言語独立に定義することで説明した. また, 部品の構築活動について検討し, Decomposition と Composition の活動を定義してこれを説明した. さらに, これらの定義を併せて構築した, 部品知識の獲得・拡張活動プロセスのモデルを提案した. これまで本章で開発してきたシステムでの説明と, 提案モデルに基づく説明を比較した評価実験では, モデルに基づいた説明について高い評価を得た.

以上の取組みを通して, 本章ではこれまで暗黙知とされていたプログラミングに関する諸要素に接近し, 学習の観点から暗黙知を形式知化した. これにより, これまで整理されていなかったプログラミングにおける知識を形式知として定義し, 知識を取り扱うためのモデルの一つを示した.

今後の課題としては, 4.7.2 で述べたことに加え, 部品知識の修正活動プロセスへの提案モデルの拡張, 提案モデルを実装したシステムの構築が挙げられる.

表 4.6 従来のシステム [古池 19a] で提示される説明・B 群におけるクラスタ分析

(a) 説明 A 群におけるクラスタ分析 (7 クラスタ)

1		2		3		4	
説明	0.353	分かる	0.583	説明 A	0.467	間違える	0.857
単純	0.286	シンプル	0.444	説明	0.429	気付く	0.857
分かる	0.188	説明 A	0.233	答え	0.185	自分	0.667
1 行目	0.143	思う	0.143	形	0.143	間違い	0.444
言葉	0.143	楽	0.143	解説	0.071	思う	0.412
短い	0.143	詳細	0.143	感じる	0.071	見る	0.333
文	0.143	人	0.143	誤り	0.071	理由	0.286
文中	0.143	無い	0.143	少ない	0.071	レベル	0.222
勉強	0.143	話	0.143	唐突	0.071	c=a,a=b,b=c	0.143
明確	0.143	解答	0.125	意味	0.067	原因	0.125

5		6		7	
答え	0.4	出来る	0.778	正解	0.25
教える	0.3	提示	0.4	思う	0.24
シンプル	0.143	次に	0.375	理解	0.235
気	0.1	答える	0.375	間違える	0.188
終わる	0.1	何故	0.222	考える	0.188
振り返る	0.1	説明 A	0.188	コード	0.143
正しい	0.1	理解	0.154	言う	0.143
答え方	0.1	if 文	0.125	今後	0.143
表示	0.1	応用	0.125	混乱	0.143
出す	0.091	覚える	0.125	実際	0.143

(b) 説明 B 群におけるクラスタ分析 (9 クラスタ)

1		2		3		4		5	
間違える	0.667	考える	0.382	丁寧	0.727	思う	0.542	細かい	0.6
合う	0.25	出来る	0.306	説明 B	0.059	理解	0.312	説明	0.103
理由	0.238	理解	0.263	理解	0.044	今後	0.262	上記	0.077
場所	0.188	答え	0.191			説明 B	0.253		
書く	0.182	教える	0.189			間違える	0.229		
箇所	0.167	提示	0.188			示す	0.171		
詳しい	0.095	仕組み	0.147			答え	0.137		
教える	0.077	プログラム	0.128			作れる	0.125		
機能	0.071	コード	0.108			プログラミング	0.119		
答え	0.063	生徒	0.094			繋がる	0.119		

6		7		8		9	
説明	0.407	必要	0.682	初心者	0.667	分かる	0.256
説明 B	0.177	処理	0.133	難しい	0.471	詳細	0.241
プログラム	0.111	直接的	0.133	学習	0.222	機能	0.2
具体	0.111	方向	0.133	分かる	0.118	誤り	0.188
間違える	0.079	教える	0.125	少々	0.118	変数	0.138
細かい	0.078	説明	0.119	複雑	0.118	構造	0.063
詳しい	0.077	機能	0.115	説明	0.107	出来る	0.049
間違い	0.073	説明 B	0.113	思う	0.097	それぞれ	0.036
状況	0.061	考える	0.071	説明 B	0.08	ヒント	0.036
作る	0.057	条件	0.067	ベスト	0.059	易い	0.036

表 4.7 理由記述文章全体におけるクラスタ分析

1		2		3		4	
細かい	0.308	必要	0.244	説明 A	0.362	間違える	0.853
説明	0.175	説明 B	0.202	順	0.191	思う	0.283
詳しい	0.121	説明	0.129	解答	0.138	気付く	0.235
状況	0.065	機能	0.086	説明	0.103	今後	0.205
行事	0.033	細かい	0.075	説明 B	0.098	間違い	0.195
使える	0.033	逆	0.046	アンケート	0.095	自分	0.163
失敗	0.033	言及	0.046	正解事例	0.095	答え	0.147
文	0.033	条件	0.046	足りる	0.087	プログラミング	0.143
要点	0.033	整う	0.046	答え	0.071	影響	0.118
流れ	0.033	構成	0.044	問題	0.067	知る	0.118

5		6		7		8	
丁寧	0.818	思う	0.313	提示	0.355	考える	0.595
説明 B	0.042	説明 B	0.248	答え	0.339	必要	0.378
理解	0.03	初心者	0.17	出来る	0.216	学習者	0.267
分かる	0.023	作れる	0.163	教える	0.161	自分	0.205
		難しい	0.156	生徒	0.136	正解	0.194
		理解	0.149	明確	0.13	問題	0.182
		正解	0.1	何と	0.095	直接的	0.179
		自分	0.083	含める	0.095	説明 A	0.178
		プログラム	0.082	最初	0.095	理解	0.158
		思い出せる	0.075	処理内容	0.095	良い	0.152

9		10		11		12	
理由	0.258	分かる	0.611	間違う	0.5	説明	0.253
書く	0.241	シンプル	0.167	合う	0.333	プログラム	0.153
出来る	0.22	説明 A	0.096	場所	0.25	具体	0.135
仕組み	0.179	大事	0.087	箇所	0.214	初心者	0.119
理解	0.162	構造	0.077	部分	0.154	機能	0.1
コード	0.132	変数	0.071	詳しい	0.118	詳細	0.095
答える	0.115	誤り	0.065	気づく	0.111	作る	0.093
説明 A	0.105	ヒント	0.046	解決	0.083	学習	0.083
何故	0.103	楽	0.046	見分ける	0.083	深い	0.082
提示	0.093	結局	0.046	元	0.083	変数	0.067

第5章 習熟度を考慮した適応的フィードバック機能

5.1 本論文における本章の位置付け

4章では、3章で提案する教授モデルを踏まえて、領域知識モデルを構築し、それによって生成しうる適応的支援の有効性を概念実証的に検証した。4章にて領域知識モデルが明らかになったことで、その領域知識に対する学習者の理解状態を表現する学習者モデルを表現できるようになった。そこで本章では、4章で構築した領域知識モデルに基づき、学習者に適応的な支援を与えるための部品の習熟度を考慮した学習者モデルを提案し、3章で示した初期のITSに対して追加実装する。これにより、プログラミングに対して、教授モデル、領域知識モデル、学習者モデルをそれぞれ構築でき、情報構造指向アプローチに基づくプログラミングを対象としたITSを実現しうることを示す。

5.2 プログラミング学習支援としての本章の動機

プログラミング学習においては、ある機能を実現するためのプログラムの構築を要求されることが多い。しかし、体系的に学ぶためには、一問一答形式で学ぶのではなく、学習したプログラムの機能を発展させ、相互のプログラムを関連付けて学ぶことが重要である。このためには、一度構築したプログラムを発展させたより高度な機能を実現するコードを構築したり、構築したプログラムの一部に機能を見出し、その一部を再利用して他の機能を実現するプログラムを構築したりすることが重要である。このような体系的な理解は、換言すれば対象のプログラムの機能の階層構造の理解、つまり機能的階層構造の理解といえる。

古池らは、この機能的階層構造の理解（古池らはこれを「構造的理解」と表記）を指向した手法および支援システムを提案・開発し、その効果を検証してきた [古池 19a]。具体的には、簡単な機能についてのプログラムを作成する課題から、作成したプログラムの再利用および拡張が必要となるような、より複雑な機能を構築する課題を段階的に提示する。この中で、一度作成したプログラムは「部品」として1行で表現できるように単純化され、学習者が再利用の利点を意識することを期待している。これにより、学習者が部分

ごとの機能とその関係性を意識しながら機能的階層構造を理解させることを狙っている。

しかし、これまで古池らのシステムでは、学習者の習熟状態を把握する枠組みはなく、それに応じた支援もない。システムからのフィードバックは、学習者の要求に応じて無制限に与えられる。学習者の学びにおいて、適応的にシステムが支援することは重要である。例えば、平嶋らは、学習者が必要な知識を持っているにも関わらず誤りを犯すケースがしばしば起こり、これを解決するためには知識の習熟状態に応じた支援が必要であることを指摘している [平嶋 92]。また、Koedinger らは、システムが過剰にフィードバックを与えることで学習者の学びが低減する危険性を指摘している [Koedinger 07]。

したがって本章では、次に述べる 2 つの仮説を立てた。平嶋らの指摘に基づき、(H1) 学習者が自身でフィードバックを得るのに比べて、適応的にフィードバックを与えるほうが高い学習効果を得られる。Koedinger らの指摘に基づき、(H2) 学習者が自身でフィードバックを得るのに比べて、適応的にフィードバックを与えるほうが学習者の受け取るフィードバックの総数^{*8}が少ない。この 2 つの仮説を検証するために、部品の習熟状態を表現する学習者モデルを取り入れ、古池らのシステムにその学習者モデルに基づいた適応的なフィードバックを実現する。さらに、上記 2 つの仮説を検証するための初期評価を実施する。

5.3 機能的階層構造の理解を指向した学習支援

5.2 節で述べたように、古池らはプログラムを段階的に拡張していくように系列的な課題を提示する「部品の段階的拡張手法」と呼ばれる学習手法を提案し、それを支援する学習支援システムを開発してきた [古池 19a]。そのインタフェースを図 5.1 に示す。

古池らによる学習支援システムでは、中央上部 (a) に問題文と問題文を解く際の条件、この問題を解くことで獲得できる部品が提示される。学習者はこの部品の獲得を目指して、システム内でプログラミングすることになる。そのために、左側 (b) に表示されているリストから選択したコードを中央 (c) の作業スペースに追加し、テキストボックスの書

^{*8}Koedinger ら [Koedinger 07] はフィードバックの頻度と学習効果の関係性について議論しているが、本章の実験ではシステム利用時間が比較的短時間 (30 分) かつ群間で統一されているため、フィードバックの頻度は総数に近似できると考えている。

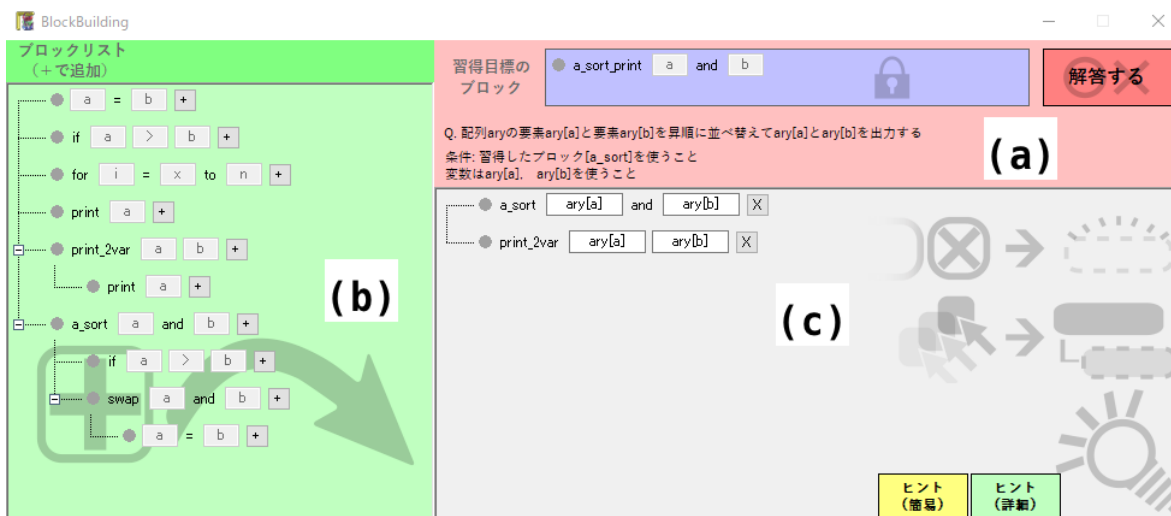


図 5.1 古池らによる学習支援システム

き換えや、順番・階層の変更を行う。この活動のなかで、学習者にフィードバックを与えるためのシステムが備えるフィードバック機能は4種類ある。1つ目は、単純ヒント機能で、学習者が組み立てているプログラム中の最も上部に存在する誤り箇所を指摘する。単純ヒントは学習者の問題解決に対して誤っている箇所を示すことで修正の手がかりを与えることを意図しており、単に学習者の誤り箇所を示唆するだけでなく、確認不足や注意不足によるいわゆる凡ミスケアする役割も担っている。2つ目は、詳細ヒント機能で、単純ヒントと同様に誤り箇所を検知し、その部分の答えを提示する。詳細ヒントは学習者の問題解決をシステムが学習者の代わりに一部修正することを意図しており、学習者が特定の箇所における問題解決に行き詰まっている場合でも、問題解決への見通しを得られることを期待している。3つ目は、正誤判定機能で、学習者のプログラムの正誤を診断する。最後に、4つ目は復習機能である。復習機能では、学習者が自身のプログラムに対して過去の問題で獲得した部品をその問題中に初めて作業スペースに追加したときに、システムはその部品を獲得した問題に移行し、復習を促す。復習では、過去の問題を振り返ることで、解法の習熟を意図している。

これらのフィードバックのなかで、単純ヒント機能と詳細ヒント機能は常に学習者の要求に応じて利用することができる。つまり、単純ヒント機能や詳細ヒント機能は無制限に

使えば答えを閲覧するのと同じ状況になってしまうため、学習者に適切に利用されるかどうかは学習者のスキルに依存する。また、復習機能は新しい問題ごとに、獲得した部品の初回利用時に（強制的に）実行される。よって、学習者にとって復習が必要ないときにもしばしば実行される可能性がある。上記の課題を解決するためには、個々の学習者の状態に適応的なフィードバックを実現できることが望ましい。そこで、本章では学習者ごとの解法（部品）の習熟状態をパラメータとした学習者モデルを提案し、従来システムに追加実装する。

5.4 部品の習熟度を用いた適応的フィードバック

学習者に対してシステムが効果的にフィードバックするためのタイミングとして、学習者が誤りを犯した際が挙げられる。しかしながら、学習者が無作為に振る舞うことで何度も誤った場合に毎回フィードバックを与えることは過剰なフィードバックが懸念される。一方、不十分なフィードバックはしばしば学習者の行き詰まりを招いてしまうことが懸念される。このような現象は「Assistance Dilemma」と呼ばれ、これを解消するためには、問題解決の難易度を適切に管理することが重要である [Koedinger 07]。また、そのためには、学習者が誤りを犯す原因を分類する必要がある。

平嶋ら [平嶋 92] は、「必要な知識を持っているにも関わらず発生する行き詰まり」および「すでに獲得している解法についての習熟」をシステムが扱えることの重要性を主張している。特に、前者の原因の一つとして、学習者が問題解決操作を行うたびに消費される有限な資源（リソース）が不足している可能性を示唆している。さらに、リソース不足においては、適切なヒントの提示や問題解決を1段階進めてあげることでリソースが増加し、問題解決の再開が期待されるとしている。そこで本章では、誤りの原因を「解法の習熟不足」「学習者のリソース不足」に分類し、また、学習者が誤りを犯した時の状態を「解法の習熟不足状態 (Insufficient Proficiency in Solution state: IPS)」と、「学習者のリソース不足状態 (Insufficient Resource of Learner state: IRL)」に切り分けて仮定することで、状態に応じたフィードバックを生成することを目指す。

学習者の解法の習熟不足を捉えるにあたっては、「ある解法（部品）を獲得するための演習において何らかの支援を得ていた場合、その解法は習熟不足である可能性が高い」と

考えられる。そこで本章では、解法の習熟を「システムから何らかの支援を一切受けずに問題を解けること」と定義した。また、ある問題の解法が導けない場合は、その解法を構成する下位解法の習熟不足である可能性が高いと考えられる。よって、IPS を、学習者の誤った問題の正解の内部に、すでに別の問題で何らかのフィードバックを受けながら獲得した解法が含まれている状態と定義した。

また、上記のリソースに関する議論から、必要な解法を習熟していたとしても問題解決に行き詰まる状況をケアする必要があるといえる。しかし、学習者のリソースを動的に推定することは困難である [平嶋 92]。そこで本章では、習熟不足の解法が無いにも関わらず誤った場合はリソース不足によるものである可能性が高いと考え、そのような状態を IRL と仮定した。したがって、IRL を、学習者の誤った問題の正解の内部にすでに別の問題で獲得した解法（獲得済み解法）が含まれていても、その獲得済み解法が一切のフィードバックを受けずに獲得されている状態と定義した。

これらの2つの誤り状態を「部品の段階的拡張手法」における部品を用いた問題解決過程に当てはめ、何らかのフィードバックを得ながら獲得した部品を IPS 部品と定義する。学習者が IPS 部品を含む解法で誤りを犯したとき、IPS と推定する。対照的に、学習者が IPS 部品を含まない解法で誤ったときには、IRL と推定する。つまり、学習者がもつ IPS 部品の有無を検知すれば、IPS か IRL かを判定できる。よって、学習者モデルは、学習者が持つ IPS 部品の集合で表現できる。

この学習者モデルを用いた診断アルゴリズムを図5.2に示す。ある問題を解くとき、システムはまず学習者の解答が間違っているかどうかを判定する。間違っていれば学習者モデルが何らかの IPS 部品を持っているかどうかを診断し、そうでなければ次の問題へ移行する。学習者モデルに IPS 部品が一切含まれない場合は、システムは学習者を IRL と推定し、詳細ヒントを与える。なぜなら、上述のように適切なヒントの提示や問題解決を1段階進めてあげることでリソースが増加し、問題解決の再開が期待されるためである。また、単純ヒントではなく詳細ヒントから与える理由は、単純ヒントを与えるのみでは学習者に示唆された誤り箇所の問題解決を要求することになり、かえってリソースの消耗が懸念されるためである。学習者モデルに IPS 部品が含まれる場合は、システムは学習者の状態を IPS と推定し、システムは学習者がまだ初回のフィードバックを受け取ってい

ないかどうかを診断する。初回のフィードバックを受け取っていない場合、学習者に単純ヒントを与える。なぜなら、初回の誤りの際は、学習者は単に注意不足や確認不足が原因で誤りに気づけなかった場合があると考えられるためである。初回のフィードバックをすでに受け取っている場合は、IPS 部品を獲得する問題の復習を促す。復習が終わった場合は、もとの問題に戻り、再度解答を行う。

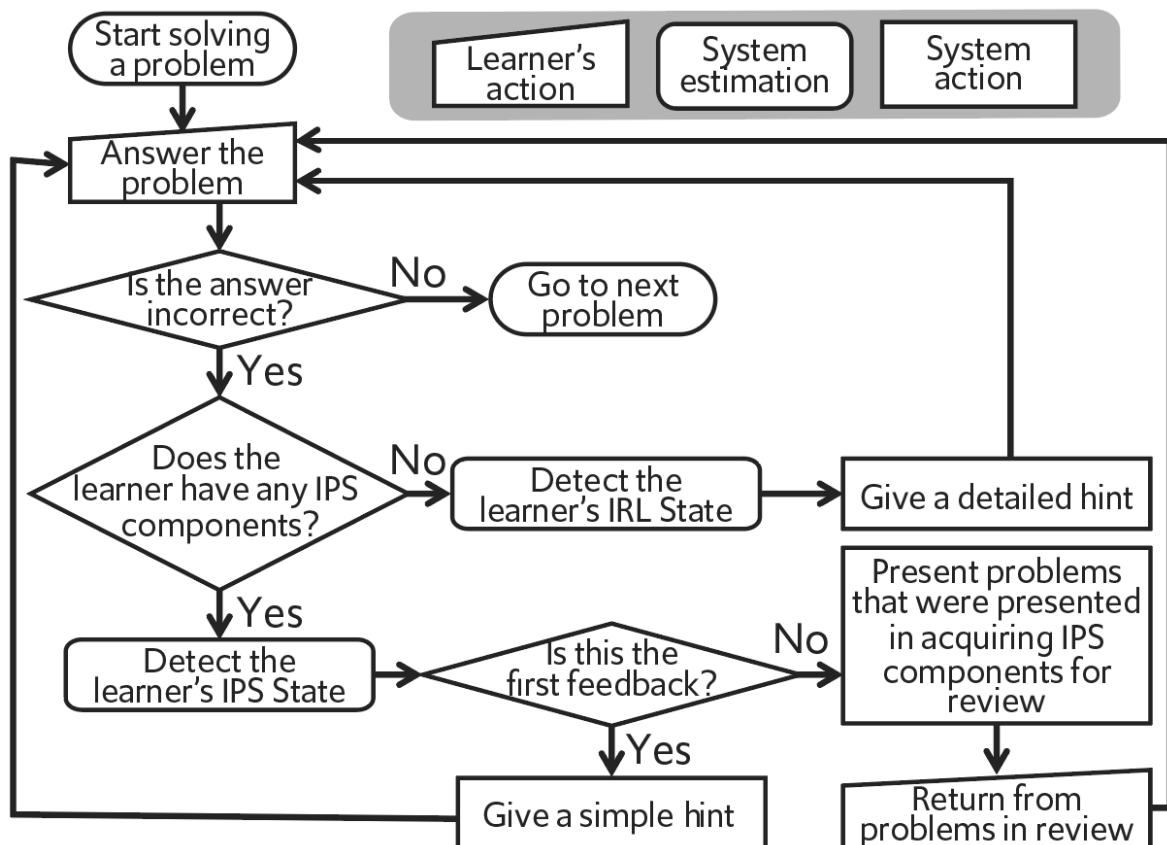


図 5.2 学習者モデルを用いた診断アルゴリズム

なお、学習者モデルにおいて、学習者が「IPS 部品を獲得した問題」の復習を完了した際に一切のフィードバックを受けていなければ、その部品は習熟状態に変化したと判断し、IPS 部品の集合として表現されている学習者モデルから、復習が完了した IPS 部品が削除される。

これらの学習者モデルと診断アルゴリズムを古池らの従来システム [古池 18a,

古池 19a] に追加実装することで、適応的フィードバックの実現を指向した。

5.5 初期評価

5.5.1 手続き

5.2 節で述べた 2 つの仮説の検証を目的に初期評価を実施した。実験条件として、以下の 2 つの条件を設けた。Adaptive Feedback (AF) 条件は、本章で提案した方法を用いたシステムを使って学習してもらう条件である。そのため、学習者が受け取るフィードバックは制限される。Optional Feedback (OF) 条件は、古池らの従来システム [古池 18a, 古池 19a] をベースに、復習も任意かつ無制限としたシステムを使って学習してもらう条件である。そのため、学習者は任意のタイミングで無制限にすべてのフィードバックを得ることができる。実験参加者は 12 名の最低 3 年以上プログラミングの講義で学習した大学生である。全ての学習者は、for や if などの基本概念、ソートアルゴリズム、関数については既習である。これらの参加者を、AF 条件で学んだ後に OF 条件で学ぶ A 群 7 名と、その反対の順番で学ぶ B 群 5 名に振り分けた。

本評価実験は、2 日間にわたって実施された。1 日目は、事前テスト (12 問) を 25 分間で実施した後、各群の 1 つ目の条件でシステム学習を 30 分間実施した。最後に、25 分間で事前テストと同じ内容で事後テストを実施し、アンケートを行った。2 日目も同様の流れで事前テストを行った後、2 つ目の条件でシステム学習を行い、事後テストとアンケートを行った。1 日目と 2 日目の、事前／事後テスト・システムにおける学習内容は全て同一のものとなっている (表 5.1)。ただし、No. 12 と 13 は No. 10 と 11 の派生系であり、重複の多い No. 12 の問題のみテスト時間の都合で提示していない。よって、抑制しようとしている順序効果は、システムへの習熟度合いだけでなく、学習内容およびテストも含むものとなっている。

各テストでは段階的に拡張できるような問題を提示しており、学習者に対しても可能な限り関数化や部品化などを用いて別の問題で解いた解法を再利用するように指示している。これらは手順点と、構造化点の 2 つで評価している。手順点は講義等で一般的に行われるように、記述された手順によって問題文の要求を満たしているか否か (1 点/問) で

表 5.1 学習範囲と事前・事後テストの内容

No.	問題文
1	変数 a と変数 b を入れ替えるプログラムを記述せよ
2	変数 a と b を出力するプログラムを記述せよ
3	変数 a と変数 b を入れ替えて出力するプログラムを記述せよ
4	変数 a が変数 b より大きいときに変数 a を出力するプログラムを記述せよ
5	変数 a と変数 b を昇順（小さい数から大きい数）に並び替えるプログラムを記述せよ
6	変数 a と変数 b を昇順（小さい数から大きい数）に並び替えて出力するプログラムを記述せよ
7	要素数 n の配列 ary を出力するプログラムを記述せよ
8	配列 ary の i から要素数 n までの最小値を ary[i] に求めるプログラムを記述せよ
9	配列 ary の i から要素数 n までの最小値を ary[i] に求めて出力するプログラムを記述せよ
10	要素数 n の配列 ary を昇順（小さい数から大きい数）に並び替えるプログラムを記述せよ（ただしアルゴリズムは単純ソートとする）
11	要素数 n の配列 ary を昇順（小さい数から大きい数）に並び替えるプログラムを記述せよ（ただしアルゴリズムはバブルソートとする）
12	要素数 n の配列 ary を昇順（小さい数から大きい数）に並び替えて出力するプログラムを記述せよ（ただしアルゴリズムは単純ソートとする）
13	要素数 n の配列 ary を昇順（小さい数から大きい数）に並び替えて出力するプログラムを記述せよ（ただしアルゴリズムはバブルソートとする）

採点している（システムで学習した解答以外も正しいければ加点している）。構造化点については、参考文献 [古池 19a] と同一の基準で、ある解法の構築に他の問題で獲得した部品を活用しているか否か（1点/問）で採点している（その解法（部品）の正しい使われ方でない場合や、正解では使われない解法（部品）であっても利用の有無により加点）。これは「学習者が部分ごとの機能とその関係性を意識し、他の問題で獲得した部品を再利用しようとしているかどうか」の評価に主眼を置いているためである。また、構造化点は何らかの部品の再利用を必要とする問題のみで評点しており、表 1 に示す No. 1 や No. 2, No. 4 といった、プリミティブな部品のみで構成される課題においては評価されない。

5.5.2 結果および考察

表 5.2 にテストにおける手順点と構造化点の結果を示す。また、実施日および採点方式（手順点・構造化）ごとに、実験参加者内要因をテストタイミング（事前・事後）、実験参加者間要因を群（A 群・B 群）とした分散分析を行った。結果を有意差（** : $p < .01$, * : $p < .05$, † : $p < .10$ ）を付記した形で示す。1 日目の手順点では、テ

表 5.2 テストにおける手順点と構造化点の結果（括弧外は平均，括弧内は標準偏差）

実施日	群	条件	手順点		構造化点	
			事前	事後	事前	事後
1 日目	A	AF	2.86 (2.10)	3.57 (2.38)	1.00 (1.60)	3.86 (2.59)
	B	OF	2.40 (1.36)	3.60 (1.20)	0.20 (0.40)	3.80 (2.64)
2 日目	A	OF	3.29 (2.60)	4.14 (3.27)	3.29 (2.49)	6.57 (2.72)
	B	AF	3.40 (0.49)	5.00 (2.28)	2.60 (1.62)	5.80 (2.32)

ストタイミング [$F(1, 10) = 5.28, p = .04^*$], 群間 [$F(1, 10) = 0.03, p = .87$], 交互作用 [$F(1, 10) = 0.34, p = .57$], 構造化点では, テストタイミング [$F(1, 10) = 20.25, p < .01^{**}$], 群間 [$F(1, 10) = 0.15, p = .71$], 交互作用 [$F(1, 10) = 0.27, p = .61$] という結果が得られた. 2 日目の手順点では, テストタイミング [$F(1, 10) = 4.88, p = .05^\dagger$], 群間 [$F(1, 10) = 0.10, p = .76$], 交互作用 [$F(1, 10) = 0.45, p = .52$], 構造化点では, テストタイミング [$F(1, 10) = 20.37, p < .01^{**}$], 群間 [$F(1, 10) = 0.29, p = .60$], 交互作用 [$F(1, 10) = 0.00, p = 1$] という結果が得られた. よって, 2 日目の手順点を除けば, どの実施日・採点方式においてもテストタイミング間に有意差がみられ, 2 日目の手順点についても有意傾向がみられた. また, 群間の有意差および交互作用はみられなかった.

表 5.3 にシステム学習時間中に条件ごとに学習者が受け取ったフィードバックの総数を示す. 同様に, 実施日ごとに, 実験参加者内要因をフィードバックタイプ (単純ヒント・詳細ヒント・復習), 実験参加者間要因を群 (A 群・B 群) とした分散分析を行った. 1 日目のフィードバックタイプでは, 群間 [$F(1, 10) = 34.34, p < .01^{**}$], フィードバックタイプ [$F(2, 20) = 25.83, p < .01^{**}$], 交互作用 [$F(2, 20) = 6.36, p < .01^{**}$] という結果が得られた. Tukey の post-hoc test を実施した結果, 群間における詳細ヒントに有意差がみられた ($p < .01^{**}$). 2 日目のフィードバックタイプでは, 群間 [$F(1, 10) = 7.35, p = .02^*$], フィードバックタイプ [$F(2, 20) = 15.37, p < .01^{**}$], 交互作用 [$F(2, 20) = 4.19, p = .03^*$] という結果が得られた. Tukey の post-hoc test を実施した結果, 群間における詳細ヒントに有意差がみられた ($p = .01^*$). これらの結果から, どちらの実施日においても, 群間とフィードバックタイプに有意差および交互作用がみられた.

よって, これらの結果は, (H1) を支持しなかった一方で, (H2) を支持している. つまり, 提案する学習者モデルを用いた適応的フィードバックにより, 学習者は部分的に答え

表 5.3 システム学習の条件ごとに学習者が受け取ったフィードバック数（括弧外は平均，括弧内は標準偏差）

			学習者が受け取ったフィードバック数		
実施日	群	条件	単純ヒント	詳細ヒント	復習
1 日目	A	AF	6.71 (2.31)	19.57 (5.73)	3.00 (1.07)
	B	OF	16.80 (13.76)	51.40 (19.44)	1.20 (1.47)
2 日目	A	OF	13.14 (3.98)	23.43 (12.01)	1.14 (1.46)
	B	AF	5.80 (2.23)	9.20 (6.34)	2.20 (1.47)

を見ることができる詳細ヒントの受け取る回数を減少させることが示唆された。よって、提案する手法によって、学習者が答えをみる回数を減らせたことが示唆された。

また、学習者の学習活動をシステム中の問題達成率の観点から調査したところ、AF 条件については A 群が 68.13% (SD: 17.44%)、B 群が 80.00% (SD: 17.71%) であり、OF 条件については A 群が 86.15% (SD: 23.33%)、B 群が 85.71% (SD: 17.44%) であった。この結果を Assistance Dilemma[Koedinger 07] の議論に照らすと、AF 条件では問題の解決速度は遅いが理解度が深まり、相対的に、OF 条件では問題の解決速度は早い但理解度が浅くなることが予想される。よって、今回の検証で H1 を支持する結果が得られなかった原因に、システムの利用時間に制限があったことにより問題達成率に差が出たことが考えられる。

また、このことに基づけば、新たな仮説として利用時間に制限がない状況で同じ問題数を解く条件においては、H1 を支持する結果が得られるのではないかと期待できる。この仮説が得られたという点も、本章の意義の一つといえる。

5.6 関連研究と本章の位置付け

本章では、仮説 (H1)、(H2) について適応的にフィードバックを得るときの効用について検討した。しかし、適応的なフィードバックの効用については、既存の ITS (Intelligent Tutoring System) 研究において多く報告されている [Bimba 17]。これに対し、本章は、問題系列を前提とした適応的フィードバックについての調査を行ったと位置付けることができる。松居ら [松居 10] は、問題系列を対象とした ITS 研究 [VanLehn 05, Graesser 99, 平嶋 95] について、問題の提示の方法を複数定義している

が、本章はある問題の解決の成功・失敗時に関連する問題を提示する問題指向型に分類される。中でも、本章は、対象とする問題系列のうち下位の問題の解法が上位の問題の解法に完全に包含されるような問題間の系列化アプローチ [平嶋 95] を採用している。同様のアプローチによって問題解決失敗時に関連する単純化した問題を与える研究 [武智 15, 相川 20] も存在するが、ある問題の提示タイミングを正誤のみで決定している。そのため、このようなアプローチにおいて、学習者の活動履歴をもとにして知識の習熟度を評価し、適応的に問題やヒントを提示する本章の成果は、一定の新規性を有すると考えられる。

5.7 本章のまとめ

本章では、機能的階層構造の理解支援における適応的なフィードバックの実現を目的に、まず、先行研究 [平嶋 92, Koedinger 07] に基づき、2つの仮説 H1 と H2 を立てた。さらに、この2つの仮説を検証するために、部品の習熟状態を表現する学習者モデルを取り入れ、古池らの既存のシステム [古池 19a] に追加実装し、初期評価を実施した。評価から得られた結果は、H1 を支持せず、H2 を支持していた。よって、適応的フィードバックを用いる条件では、学習者が自由にフィードバックを得られる条件よりも答えをみる回数を減らせることが示唆された。

また、今回は条件間に学習効果における有意な差がみられなかった。よって、適応的フィードバックにより答えをみる回数を減らした場合に、同等の学習効果が得られる可能性もあるが、標本数を増やしたより精緻な実験が必要である。よって今後は、学習効果の追証や学習ログを用いた詳細な分析が課題である。

第II部

領域知識モデルの 汎用的な再利用法の実現

第 6 章 誤りの可視化を重視した Teachable Agent の挙動制御手法

6.1 本論文における本章の位置付け

本論文では情報構造指向アプローチに基づく ITS の汎化を目指し、第 II 部では、情報構造指向アプローチに基づく ITS が未知の領域にも適用しうる発展を実現した。しかし、2 章で指摘したような、ITS1 つあたりの開発コストは依然として高いままとされている。そこで第 III 部では 2 つ目の課題として、情報構造指向アプローチに基づく既存の ITS から、領域知識モデルを取り出し、別の教授モデルに対して再利用しうることを示すことを目的としている。

本章では、情報構造指向アプローチに基づく領域知識モデルを汎用的に再利用可能な Learning-by-teaching を対象とした教授モデルを提案し、実際にその教授モデルを用いて既存の領域知識モデルが再利用しうることを示す。これにより、情報構造指向アプローチに基づく領域知識モデルが実際に他の教授モデルに適用できること、また、どのような教授モデルであれば、また、どのような領域知識モデルであれば汎用的に互いを組み合わせられるかを明らかにする。

6.2 Teachable Agent を用いた学習支援としての本章の動機

学習者が理解を深めるための代表的な手段の一つとして他者への説明がある。このような行為を応用した学習手法に、教えることによる学び、すなわち Learning-by-teaching (LbT) がある [Duran 17]。LbT では、他者に学習内容を教示することで、自身の理解を深められることが知られている [Fiorella 13]。

LbT の実現には、教えられる側である他者の知識や教示を受ける際の態度といった「教えられる側の振舞い」や、「教えられる側の存在」そのものが重要となる。例えば、自身と他者の知識が同程度であったり、もしくは、自身と他者の知識が大きくかけ離れている場合には、その間の教示は困難になる。同様に、他者がそもそも教示を受け入れる態度ではない場合なども、LbT の実現が困難となる。さらに、教室や講義での実施を考えた場合、

LbT が実現できるのは「教える側」のみとなり、最小構成が 2 人一組であることを考えると、半数以上が LbT を実行できないことになる。

「教えられる側の振舞い」の制御および「教えられる側の存在」の担保の 2 つの課題を同時に解決できる技術として、Teachable Agent (TA) がある [Biswas 05, Leelawong 08, Matsuda 13, 松田 18]。TA とは、学習者に教えられたことを学んでいるように振る舞うエージェントおよびその技術の一つである。この TA によって、PC やタブレットなどコンピュータ上に「教えられる側の存在」が担保されることで、すべての学習者が LbT を実行できる。また、TA を学習支援システムと統合することで、TA は学習者の知識レベルを取得することができる。これにより、学習者の知識レベルを踏まえた TA の挙動として (1) 学習者の現状の知識レベルで教えられるような知識を備えたり、(2) 現状の知識レベルから段階的により知識を深められるような振舞い、を実現できる。

こうした TA の実体は学習支援システムに付随する単なるメタファーであるにもかかわらず、学習者に対して自身を共同学習者であると認識させる「社会的スキーマ」をもたらすことが知られている [Biswas 05, Chase 09]。社会的スキーマの導入による効果は、様々な形で認められてきた [Chase 09, Gulz 20, 松田 18, Tärning 19, Okita 13]。特に、TA がもたらす「再帰的フィードバック (Recursive Feedback)」が、学びに有効であることが示されている [Okita 13]。再帰的フィードバックとは、学習者自身が学んだことを他者に教え、その結果他者がどのように振る舞うかを観察する、という過程によって発生する再帰的なフィードバックである。

再帰的フィードバックによる学習効果は、社会的スキーマという側面によるものだけではなく、TA が「学習者による仮説検証のためのシミュレータ」の役割を有することによるものも含まれると考えられる。後者に基づく学習支援を指向した研究に、「誤りの可視化」に関する一連の研究がある [Hirashima 98, Horiguchi 14]。誤りの可視化とは、学習者の誤った考えや誤答が妥当ではない結果を導くことをシミュレーションを通じて示し、学習者の誤りへの気づきを誘発する学習アプローチである。誤りの可視化では、理解させたい条件を制約としてあらかじめ定義し、学習者の解答を制約に基づいて診断する。そして、学習者の解答において充足する、もしくは違反する制約に沿ってシミュレーションを可視化し、学習者による誤りへの気づきをねらう。

これまでの TA による再帰的フィードバックは、例えば、学習者の教示から機械学習手法によって学んだ結果としての確率的な振舞い [Matsuda 13, 松田 18] や、学習者が与えたコンセプトマップに基づいてクイズに答えた結果の振舞い [Biswas 05, Leelawong 08] などであった。前者は認知的な忠実性がある反面、不確定要素が含まれる。また、後者では、学習者の知識構造が柔軟に表現できる一方で、教示するコンセプトマップの規模は実質無限である。よって、どちらの設計指針においても、学習者が TA の再帰的フィードバックから自身の教示内容の誤りを探索および発見するためには、一定の難しさが懸念される。つまり、学習者が再帰的フィードバックに基づいて対象の学習課題における制約に気づくことが困難になる。そこで、本章では制約の理解を指向した再帰的フィードバックの実現を目指し、「学習者による仮説検証のためのシミュレータ」としての忠実さを重視した TA の挙動制御手法 TAME (Teachable Agent Module for Error-visualization) を提案する。これは、誤りの可視化技術を TA の再帰的フィードバックに応用することで、制約に基づいて誤りの可視化を行う TA の挙動制御を実現する手法である。この手法により設計された挙動制御モジュールを持つ TA は、あらかじめ規定されたいくつかの制約に基づいて、制約の充足・違反の 2 種類の組み合わせからなる有限個のパターンから学習者に再帰的フィードバックを提供する。さらに、ケーススタディとして、学習支援システムの一つである「モンサクン」[前田 15, 山元 17, 平嶋 19] をベースに、提案手法である TAME を用いた TA を実現し、その評価を行う。

6.3 関連研究

6.3.1 Teachable Agent

通常、TA は、何らかの学習課題を持った学習支援システムに付随する。例えば、Matsuda らの SimStudent は、APLUS と呼ばれる知的学習支援システムのなかで、学習者と共に学んでいるように振る舞うことができる TA である [Matsuda 13, 松田 18]。この枠組みでは、SimStudent は APLUS の知識を理解していないように振る舞うことで、APLUS から学んだ学習者による教示を促す。そして、学習者から教えられた知識を、SimStudent は機械学習手法を用いて学ぶ。この手法の採用により、SimStudent は通常

の学習者と同様の学び方をしているように見せる，すなわち認知的忠実性を担保している．他の例として，Biswas らの Betty's Brain は，教育の経験や領域に関する知識がほとんど無い学生でも使用できるようにするために，エージェントの思考や推論のプロセスを予測可能になるような設計を前提としている [Biswas 05, Leelawong 08]．具体的には，事例から学んだり帰納法で学ぶことはせず，学習者からコンセプトマップを用いて教わったことだけを学び，それを誤りの理由にする．

TA がもたらす社会的スキーマの影響は大きい．例えば，Chase らは，社会的スキーマが保護者効果 (Protégé Effect) をもたらすことを明らかにしている [Chase 09]．保護者効果とは，学習者がシステム上の TA を被保護者 (Protégé) と認識することで，チューターとしての責任を果たそうとするようになる効果である．この効果により，学習者は「自身が責任を持ってチューティ (TA) を教えないといけない」という責任感から，自分自身のために学ぶ時よりも努力して学ぶようになることが知られている．Okita らは，学習者自身が学んだことを TA に教えるという形で外在化し，その結果 TA がどのように振る舞うかを観察することによる「再帰的フィードバック (Recursive Feedback)」が，学習者の学びに有効であることを示している [Okita 13]．具体的には，TA の「実験参加者が TA に教えた後，TA が教えられた内容を用いてどのように振る舞うか」を「観察する群」と「観察しない群」に分け実験を行い，前者が有意に学習効果が高かったことをもって TA が学習者に対して「再帰的フィードバック」をもたらすことを示した [Okita 13]．また，社会的スキーマの導入は学習者の視線制御 [Gulz 20] や自己肯定感 [Tärning 19] といった認知状態にも影響することがわかっている．つまり，TA が学習者にとって社会的存在であると認められることで，このような様々な恩恵をもたらすのである．

6.3.2 誤りの可視化

「誤りの可視化」とは，学習者の誤答に基づいたおかしな挙動をシミュレーションとして生成することにより，学習者自身の誤りに気付かせることで学びを誘発する学習アプローチである [Hirashima 98, Horiguchi 14]．学習者は自身の考えに基づく仮説を入力し，シミュレーションにより正誤を検証する試行錯誤を繰り返すこととなる．

誤りの可視化においては，制約 (学習題材において学ぶべきルール) の存在が重要とな

る。学習者の解答がどのような制約を充足もしくは違反しているかによって、異なるシミュレーションが生成される。例えば、初等力学を対象とした誤りの可視化を例に挙げて説明する。「物体が床の上に静止しているとき、働いてる力を作図して答えなさい」という問題のとき、満たすべき制約は、「重力が働いている」「垂直抗力が働いている」の2つである。このとき、学習者が重力のみを作図して解答した場合、前者の制約を充足し、後者の制約を違反することになる。結果として、システムは物体が床に沈むようなシミュレーションを示す。

誤りの可視化を用いた学習支援は、物理をはじめ、さまざまな範囲で実現され、有効性が示されてきた。

6.3.3 先行研究における課題と本章の位置付け

TA において再帰的フィードバックの効果に重点を置いたとき、既往研究の TA の挙動制御手法では、学習者が TA から得られた応答から自身の誤りを推測できるように TA を制御することは難しい。例えば、SimStudent[Matsuda 13, 松田 18] では機械学習的手法を採用しているため、TA は確率的に振る舞うことになる。この場合、TA は徐々に知識を学ぶような挙動を振る舞うため、必ずしも学習者の直前の入力（解答）が直後の TA の振舞い（すなわち、再帰的フィードバック）に反映されない。これにより、「何を教えたから TA が間違ったのか」を学習者が推測することが困難になるといえる。また、Betty's Brain[Biswas 05, Leelawong 08] の手法は、学習者に作成させたコンセプトマップに基づいて TA が推論することで挙動を生成する。つまり、最小のプリミティブを定義することによる知識モデリング手法 (c.f., [溝口 95]) を採用しており、全ての挙動が解釈可能なように設計されている。しかし、最小のプリミティブを無制限に組み合わせることが可能となっているため、学習者の解答が複雑になるにつれて、必ずしも TA から得られた応答が学習者の誤りを示唆できるとは限らない。

つまり、「何を教えたから TA が間違えたのか」を推測可能とするためには、学習者の直前の入力（解答）が直後の TA の振舞い（再帰的フィードバック）に反映されないという課題 (1) と、学習者の入力（解答）およびそれに対応する TA の振舞い（再帰的フィードバック）が、学習者の教示内容の誤りを示唆していることが保証されないという課題 (2)

を解決する必要がある。これらの課題の解決は、本章の目的である「制約の理解を指向した再帰的フィードバック」の実現には不可欠である。そこで、本章では、この2点の課題にアプローチするための手法を提案する。

6.4 提案手法

6.4.1 概要

ユーザ（学習者）と同様に、TA もユーザとして同様の学習対象を学習したという文脈を持つ（SimStudent[Matsuda 13, 松田 18] や Betty’s Brain[Biswas 05, Leelawong 08] はその典型例である）。よって、学習者はシステムによる学習ののち、自身と同様に学んだものの十分に理解できていない（ように振る舞う）TA に対して LbT を実施する。

そこで、本章では、「問題演習フェーズ」と TA に対して「教えるフェーズ」の2段階の学習フェーズを実現するための TA の制御機構 TAME を提案する。本機構は既存の学習支援システムに追加実装する形となる。TAME の全体像を図 6.1 に示す。TAME において、学習者ははじめに通常の問題演習を行うことを想定している。その際、TAME は学習者の問題演習時の学習履歴を取得する。TAME は学習者の学習履歴に応じて、学習者と

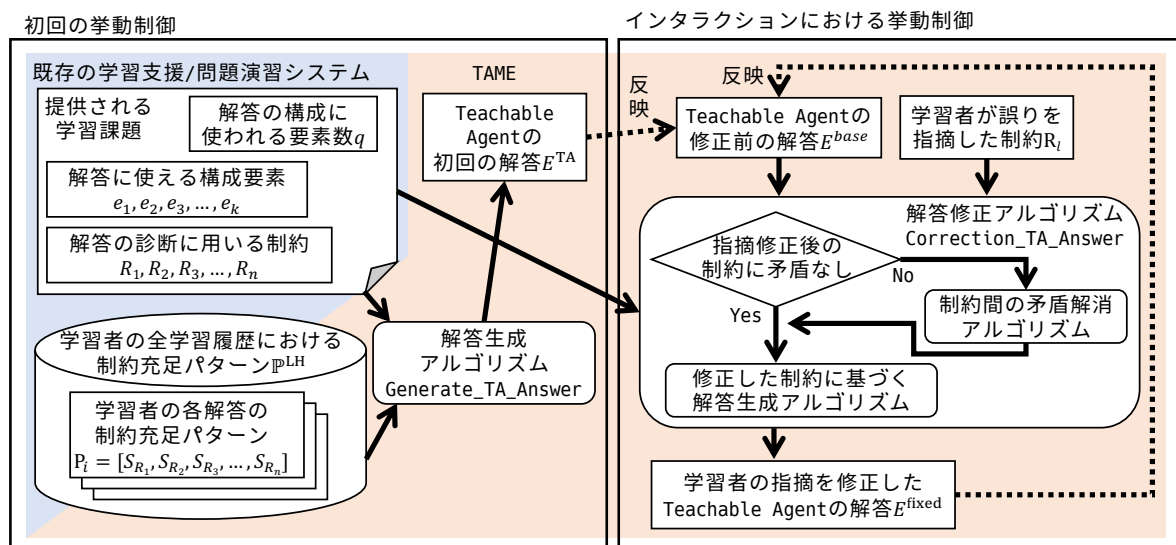


図 6.1 提案手法の全体像

類似した誤りを有する初回の挙動を生成する。次に、学習者は教えるフェーズにて TAME の挙動に応じて、TAME が誤っているであろう知識（ここでは関係性としての制約）について指摘する。指摘を受け、TAME は学習者に教えられた制約に基づく挙動を修正する。つまり、TAME は学習者から指摘された制約を修正し、修正した制約に基づいて制約充足問題を解くことで挙動を修正する。TAME が正しい挙動（正解）を示すようになったとき教える活動は終了となり、次の問題演習へと移行する。6.5 のケーススタディに照らし合わせると、既存の学習支援システムであるモンサクン [前田 15, 山元 17, 平嶋 19] において問題解決を行わせたのち、その解答履歴に応じてエージェントは初回挙動を生成し、学習者はモンサクンにおける制約に応じて、制約を教える活動を行うこととなる。

6.4.2 制約

TAME では、最終的に制約充足問題として解くという性質上、学習者の解答において満たすべき制約を定義する必要がある。この制約とは、解答中の要素が満たすべきルールであり、同時に、解答そのものは要素の組み合わせとして表現できる必要がある。つまり、定義したすべての制約を満たすときに正答となるように定義し、どれか一つでも満たさない場合には誤答になるように定義する必要がある。したがって、制約とは、対象とする学習活動において学ばせたい概念そのものと言い換えることができる。例えば、簡単な例として、仮言三段論法において「(a) ならば (b), (c) ならば (d), したがって, (e) ならば (f)」(文中の括弧はそれぞれ空欄) という問題を設定したとき、満たすべき制約の一例は、「 R_1 : (a) と (e) の一致」「 R_2 : (b) と (c) の一致」「 R_3 : (d) と (f) の一致」となる。このように制約 R_1 – R_3 を規定すれば、すべての制約を満たすと正答になり、どれか一つでも満たさなければ誤答になる。なお、制約間は独立であることが望ましいが、重複や階層性がある場合でも TAME においては問題ない。例えば、空欄に当てはまる全構成要素 e^{all} として、「 e_1 : ソクラテス」「 e_2 : ソクラテス」「 e_3 : 死ぬ」「 e_4 : 死ぬ」「 e_5 : 人間」「 e_6 : 人間」を用意したとする。この場合、複数ある正答の一つとしては、「(e_2 : ソクラテス) ならば (e_5 : 人間), (e_6 : 人間) ならば (e_4 : 死ぬ), したがって, (e_1 : ソクラテス) ならば (e_3 : 死ぬ)」となる。解答に使える全要素の個数 k は 6 個、解答に使える要素の数 q も (a) – (f) の 6 個であり、解答の組み合わせ方に順序が関係するため、ありうる

解答の全組み合わせ \mathbb{E} は ${}_6P_6 = 720$ 通りである。よって TAME では、「(a) と (e) の一致」の制約に違反した解答の一つとしてこの 720 通りを元に、「(e_4 : 死ぬ) ならば (e_6 : 人間), (e_5 : 人間) ならば (e_3 : 死ぬ), したがって, (e_2 : ソクラテス) ならば (e_1 : ソクラテス)」が生成できる。

このような制約の定義は、通常の学習支援システムでも活用されてきている。例えば、算数文章題の作問課題 (e.g., $30 \times 4 = 120$ の式で解ける問題を作りなさい) では、複数の正答がありえるため、解答に対して教師が個々にフィードバックをするのは負荷が高い。そこで、本章のケーススタディとして選定したモンサクン [前田 15, 山元 17, 平嶋 19] では、カードの組み合わせとして解答方式を構造化することで、制約を用いて解答の診断を行う。具体的には、「りんごが一個あたり 30g あります」「りんごが 4 個あります」「りんごが 120g あります」といった正答の組み合わせに必要なカード 3 枚と、「みかんが 4 個あります」「りんごが一個あたり 60g あります」といったダミーのカード 2 枚があるとき、学習者の解答が「りんごが一個あたり 30g あります」「みかんが 4 個あります」「りんごが 120g あります」といった組み合わせであれば、「文章内のもの (りんごやみかんなど) が正しい関係性である」という制約に違反している、というように診断している。TAME では、このような制約を応用して、「文章内のもの (りんごやみかんなど) が正しい関係性である」という制約の理解だけが欠落した TA の挙動として、「りんごが一個あたり 30g あります」「みかんが 4 個あります」「りんごが 120g あります」といった解答を生成する。

6.4.3 初回の挙動制御

学習支援システムでは学習者は問題解決を行い、誤答を繰り返したのちに正答に至る想定を持つ。この際、TAME では学習者が繰り返した誤答に基づき TA の初回の挙動を制御する。そこで本節は、図 6.1 の左部と対応し、具体的な初回の挙動制御について述べる。

アルゴリズム 1 では、学習者の全学習履歴における制約充足パターン \mathbb{P}^{LH} と、解答の組み合わせに使える全要素 e^{all} 、解答の組み合わせに使える要素数 q を用いることで、TA の初期の解答として設定する要素の組み合わせ E^{TA} を得ることができる関数 $\text{GENERATE_TA_ANSWER}(\mathbb{P}^{\text{LH}}, e^{\text{all}}, q)$ について記述している。例えば、制約が 6.4.2 で触れた仮言三段論法の制約 R_1 – R_3 である場合、 R_1 のみに違反する解答の制約充足パ

ターン P は $[0,1,1]$ となる. よって, 学習者の学習履歴における制約充足パターンは, $\mathbb{P}^{\text{LH}} = [010$ (初回の解答), $001, 110, 100, 110, 000, 110, 111$ (正解)] のようになる. 次に, 2行目では, $P^{\text{most_violated}}$ に \mathbb{P}^{LH} のうち最も出現頻度が高いパターンを代入する*⁹. 上

*⁹学習者の解答履歴に制約違反を含むパターンが存在しない場合, すべての制約に違反するパターンの解答を誤答

アルゴリズム 1 解答履歴に基づく TA の挙動生成

Input:

\mathbb{P}^{LH} : 学習者の全学習履歴における制約充足パターン
 e^{all} : 解答の組み合わせに使える全要素 ($e^{\text{all}} = [e_1, e_2, e_3, \dots, e_k]$)
 q : 解答の組み合わせに使われる要素数

Parameters:

P : 制約充足パターン ($P = [S_{R_1}, S_{R_2}, S_{R_3}, \dots, S_{R_n}], R$: 制約, n : 制約の個数)

S : 制約の充足状態 ($S = \begin{cases} 0: \text{制約に違反} \\ 1: \text{制約を充足} \end{cases}$)

\mathbb{P} : 各解答の組み合わせに対応した全制約充足パターン
($\mathbb{P} = [P_1, P_2, P_3, \dots, P_m], m$: 解答の組み合わせの総数)

\mathbb{E} : 対象の学習活動で実現できる要素の全組み合わせ

($\mathbb{E} = [E_1, E_2, E_3, \dots, E_j],$
 $j = \begin{cases} k P_q: \text{要素の組み合わせ方に順序が関係する} \\ k C_q: \text{要素の組み合わせ方に順序は関係しない} \end{cases}$)

E : 対象の学習課題における要素の組み合わせからなる解答 ($E = [e_1, e_2, e_3, \dots, e_q]$)

e : 解答に用いられる要素

k : 全要素の個数

Output:

E^{TA} : TA の初期の解答として設定する要素の組み合わせ

```

1: function GENERATE_TA_ANSWER( $\mathbb{P}^{\text{LH}}, e^{\text{all}}, q$ )
2:    $P^{\text{most\_violated}} \leftarrow \text{MOST\_FREQUENTLY\_PATTERN}(\mathbb{P}^{\text{LH}})$ 
    $\triangleright \mathbb{P}^{\text{LH}}$  から最頻出のパターンを抽出する処理として略記
3:    $\mathbb{E} \leftarrow \text{ALL\_COMBINATION}(e^{\text{all}}, q)$ 
    $\triangleright e^{\text{all}}, q$  を用いて想定しうる解答  $E$  の全組み合わせを  $\mathbb{E}$  に格納する処理として略記
4:    $\mathbb{E}^{\text{match}} \leftarrow \text{COMPLIANT\_PATTERNS}(P^{\text{most\_violated}}, \mathbb{E})$ 
5:    $E^{\text{TA}} \leftarrow \text{RANDOMIZED}(\mathbb{E}^{\text{match}})[0]$ 
    $\triangleright$  リストをランダムに並び替え, その後 1 つ目を取り出す処理として略記
6:   return  $E^{\text{TA}}$ 
7: end function
8: function COMPLIANT_PATTERNS( $P^{\text{query}}, \mathbb{E}$ )
9:   for each  $E_i \in \mathbb{E}$  do
10:     $P_i \leftarrow \text{SATISFYING\_CONSTRAINT\_PATTERN}(E_i)$ 
     $\triangleright$  解答  $E$  における制約充足パターン  $P$  を得る処理として略記
11:    if  $P_i = P^{\text{query}}$  then
12:       $\mathbb{E}^{\text{match}} \leftarrow \mathbb{E}^{\text{match}} + E_i$ 
13:    end if
14:  end for
15:  return  $\mathbb{E}^{\text{match}}$ 
16: end function

```

記の学習履歴の場合、110 が最頻出パターンとして特定される。このとき、単体の制約の違反で最も頻出したものではなく、制約組み合わせの違反で最も頻出したものを特定する点に注意が必要である。学習者のある誤答は1つ以上の制約違反により構成されているため、学習者の誤りを可視化するには制約違反単体ではなく、制約違反の組み合わせが必要となるためである。なお、アルゴリズム中では示していないが、最も出現頻度の高いパターンが一意に定まらない場合は、その中でいずれか一つを選択する。3行目では、 \mathbb{E} に想定しうる全ての解答の組み合わせを生成し、格納する。4行目では、8–16行目の関数によって、 \mathbb{E} の各要素の制約パターンを一つずつ得ながら、制約が $P^{\text{most_violated}}$ と一致するものだけを $\mathbb{E}^{\text{match}}$ に格納する。このように、特定した制約充足パターンに基づいて、エージェントは対象の学習活動で実現できる要素の組み合わせを用いて制約充足問題を解く。5行目では、解答の組み合わせのリスト $\mathbb{E}^{\text{match}}$ を無作為に並び替え、その中から1つ目を E^{TA} に代入する処理をしている。これは、同じ制約充足パターンであれば、解答は同じ誤り方をしており、本章の目的である「制約の理解を指向した再帰的フィードバック」という観点からは同質であることが保証されるためである。最後に6行目で E^{TA} の値を返して終了となる。こうして、学習者と同じ制約の誤り方に基づいた誤答が生成される（必ずしもまったく同じ誤答が生成されるわけではない）。つまり、この時点で学習者の知識状態を反映した誤りの可視化をしているといえる。

6.4.4 インタラクションにおける挙動制御

本節では図6.1の右部と対応し、教えるフェーズ中のインタラクションにおける挙動制御について述べる。TAが初回の挙動制御により示した誤答をもとに学習者はLbTを行う。

学習者は、初回の挙動制御によりTAが示した誤答をもとにLbTを行う。よってTAの解答 E^{TA} から学習者の指摘した制約 R_l を修正し、修正された制約に基づいて解答を生成する流れをアルゴリズム2として示す^{*10}。アルゴリズム2では、TA

として提示する。これは、このような既に誤りなく正解できた学習者に対して、より制約に対する学びを深めてもらうことを意図しているためである。

^{*10}アルゴリズム1と同じ記号や関数の説明は紙面の都合上省略

アルゴリズム 2 指摘に基づく TA の解答修正

Input:

E^{base} : TA の修正前の解答

R_l : 学習者が TA の修正前の解答に対して誤りを指摘した制約

e^{all} : 解答の組み合わせに使える全要素 ($e^{\text{all}} = [e_1, e_2, e_3, \dots, e_k]$)

q : 解答の組み合わせに使われる要素数

Output:

E^{fixed} : TA の修正後の解答

```
1: function CORRECTION_TA_ANSWER( $E^{\text{base}}, l, e^{\text{all}}, q$ )
2:    $\mathbb{E} \leftarrow \text{ALL\_COMBINATION}(e^{\text{all}}, q)$ 
3:    $P^{\text{base}} \leftarrow \text{SATISFYING\_CONSTRAINT\_PATTERN}(E^{\text{base}})$ 
4:    $P^{\text{suggest}} \leftarrow \text{FIXED\_CONSTRAINT\_PATTERN}(P^{\text{base}}, R_l)$ 
5:    $\mathbb{E}^{\text{match}} \leftarrow \text{COMPLIANT\_PATTERNS}(P^{\text{suggest}}, \mathbb{E})$ 
6:   if LENGTH( $\mathbb{E}^{\text{match}}$ ) = 0 then
7:      $P^{\text{resolved}} \leftarrow \text{CONFLICT\_RESOLVED\_PATTERN}(P^{\text{base}}, P^{\text{suggest}}, R_l, \mathbb{E})$ 
8:      $\mathbb{E}^{\text{match}} \leftarrow \text{COMPLIANT\_PATTERNS}(P^{\text{resolved}}, \mathbb{E})$ 
9:   end if
10:   $E^{\text{fixed}} \leftarrow \text{RANDOMIZED}(\mathbb{E}^{\text{match}})[0]$ 
11:  return  $E^{\text{fixed}}$ 
12: end function
13: function FIXED_CONSTRAINT_PATTERN( $P^{\text{base}}, R_l$ )
14:   $P^{\text{suggest}} \leftarrow P^{\text{base}}$ 
15:   $P^{\text{suggest}}[R_l] \leftarrow \neg(P^{\text{suggest}}[R_l])$ 
16:  return  $P^{\text{suggest}}$ 
17: end function
18: function CONFLICT_RESOLVED_PATTERN( $P^{\text{base}}, P^{\text{query}}, R_l, \mathbb{E}$ )
19:  for each  $E_i \in \mathbb{E}$  do
20:     $P_i \leftarrow \text{SATISFYING\_CONSTRAINT\_PATTERN}(E_i)$ 
21:    if  $P_i[R_l] = P^{\text{query}}[R_l]$  then
22:       $\mathbb{E}^{\text{temp}} \leftarrow \mathbb{E}^{\text{temp}} + E_i$ 
23:    end if
24:  end for
25:   $\mathbb{P}^{\text{diff\_sorted}} \leftarrow \text{RANDOMIZED}(\mathbb{E}^{\text{temp}})$ 
  order by COUNT_OF_DIFFERENT_CONSTRAINT( $P^{\text{base}}, P_i \in \mathbb{P}^{\text{temp}}$ )
  ▷ ランダムに並び替えられた  $\mathbb{P}^{\text{temp}}$  を、さらに「 $P^{\text{base}}$  と比較して異なる制約の数」の昇順に並び替える
  処理として略記
26:   $P^{\text{resolved}} \leftarrow \mathbb{P}^{\text{diff\_sorted}}[0]$ 
27:  return  $P^{\text{resolved}}$ 
28: end function
```

が示した修正前の解答 E^{base} と、学習者がその解答に対して指摘した制約 R_l 、解答の組み合わせに使える全要素 e^{all} 、解答の組み合わせに使える要素数 q を用いることで、学習者の指摘を修正後の TA の解答の組み合わせ E^{fixed} を得ることができる関数 $\text{CORRECTION_TA_ANSWER}(E^{\text{base}}, l, e^{\text{all}}, q)$ について記述している。また、学習者が「TA の（指摘を修正する前の）解答 E^{base} は R_3 を満たしてない」と指摘した場合は、

$R_l = R_3$ となる。2行目では、アルゴリズム 1 と同様、 \mathbb{E} に想定しうる全ての解答の組み合わせを生成し、格納している。3行目では、まず、 E^{base} の制約充足パターンを診断し、 P^{base} に代入する。例えば、 E^{base} が制約 R_1 と R_3 を満たすときには、 $P^{\text{base}} = [1, 0, 1]$ のようになる。次に4行目では、13–17行目からなる関数で、 P^{base} から学習者の指摘を修正した制約充足パターン P^{suggest} を得る。例えば、 $R_l = R_3$ で、 $P^{\text{base}} = [1, 0, 1]$ ならば、 $P^{\text{suggest}} = [1, 0, 0]$ となる。次に5行目では、アルゴリズム 1 と同様に、制約が P^{suggest} と一致する解答だけを $\mathbb{E}^{\text{match}}$ に格納している。6行目で、 $\mathbb{E}^{\text{match}}$ が空かどうかを確認して、空でなければ（1つでも制約が P^{suggest} と一致する解答があれば）、アルゴリズム 1 と同様、10–11行目でその中の1つを選んで TA の修正後の解答 E^{fixed} とする。 $\mathbb{E}^{\text{match}}$ が空だった場合、 P^{suggest} と制約充足パターンが一致する解答が存在しない、すなわち制約間に矛盾が発生している状態なので、矛盾を何らかの形で解消する必要がある。TAME では、この解消を、学習者が直前に指摘した制約の修正を優先する形で行う。つまり、学習者が直前に指摘した制約を修正しても成立する解答を導き出すために、その他の制約の充足状態に変更を加える。よって7行目では、18–28行目の関数において、 P^{suggest} の矛盾を解消するために、学習者の指摘 R_l を修正しつつ、修正前の解答における制約パターン P^{base} に最も近い（制約間の差分が少ない）制約パターンを P^{resolved} として抽出する。最も近い制約パターンが同率で複数出現（例えば、110 と 101）した場合は、いずれかのパターンを無作為に選出する。これは、110 と 101 は、同率の優先順位であり、どちらも同程度に学習者の指摘を反映したものであると考えているためである。その後、8行目で5行目と同様に制約が P^{resolved} と一致する解答だけを $\mathbb{E}^{\text{match}}$ に格納し、10–11行目でその中の1つを選んで TA の修正後の解答 E^{fixed} とする。以降、教えるフェーズでは、学習者が指摘した制約に基づいて TA の理解状況（制約充足パターン）を修正し、その都度、その理解状況に基づく解答を提示することで誤りの可視化を行い、学習者による制約充足パターンの修正を促すインタラクティブな学習を実現する。

TAME では、学習者の入力を「TA が示した解答に内在する制約充足パターン」における制約違反の指摘とすることで、指摘箇所を優先的に修正した制約充足パターンを生成する。さらに、その制約充足パターンに基づいて解答を生成することで、必ず学習者が指摘した箇所（制約）についての修正がなされた TA の解答が生成されることになる。これに

より、6.3.3 の課題 1 に対して解決を図っている。また、課題 2 に対しては、学習課題に応じてあらかじめ規定されたいくつかの有意義な制約に基づいて、TA の挙動が各制約の充足・違反の 2 種類の組み合わせからなる有限個のパターンに類型されるよう制御している。先述の課題 1 への対応と組み合わせると、このような制御によって、TA の挙動は必ず学習者の指摘に対応した制約の充足パターンを満たすことになる。したがって、学習者が TA の挙動から指摘した制約がどのような振舞いをもたらすのかを観察できることが、提案手法で保証される。

なお、本提案手法は、既存の学習支援システムに追加実装されることを前提としているため、制約の有意義さについては保証しない。今回のケーススタディで選定したモンサクン [前田 15, 山元 17, 平嶋 19] についても、既に先行研究で制約の有用性が実証されている。

6.4.5 対象範囲

TAME が適用可能なドメインは、(条件 1) 学習活動が構成要素の組み立てとして設計されており、(条件 2) 構成要素間の関係性が制約として記述されている、という範囲である。本章では算数の文章題を対象としたモンサクンがこれを満たした事例として紹介したが、6.4.2 で述べた三段論法の範囲としては北村ら [北村 17] による論理構造の組み立て演習を対象とした学習支援システムも対象にできる。その他、力学においても、力の作図を構成要素とした課題では、「作用と反作用は釣り合う」「静止している系の任意の方向の力は釣り合う」などの制約を記述可能であり、6.3.2 で挙げた Hirashima ら [Hirashima 98] や Horiguchi ら [Horiguchi 14] の研究も対象である。よって、上記のような条件 1, 2 を満たすドメインに対して適用可能である。

6.5 ケーススタディ

6.5.1 概要

ケーススタディとして、学習支援システムであるモンサクン [前田 15, 山元 17, 平嶋 19] の学習活動を対象に TAME を適用できるかどうかの概念的な実証と、モンサクンの枠組みに対して TA の追加実装と評価を行うことで TAME の実現可能性と有効性を評価する。

6.4.2 で述べたように、モンサクンは算数文章題の作問学習を対象としたシステムである。モンサクンでは、選択肢としてカードが提示されるので、その組み合わせを用いて解答を行う。これまで加減算や乗除算などの算数文章題の作問学習がモンサクンにおいて支援されてきたが、今回は乗除算を対象とした。その理由は、理解すべき制約の数が加減算は5つだが、乗除算は7つあり、より制約の理解が重要であるためである。

6.5.2 乗除算の作問支援における制約

前田ら [前田 15] は、山元ら [山元 17] のシステムに用いられている乗除算における問題の診断方法について説明している。この診断では、次の7つの要素が活用されている。すなわち、(1) 文章構造、(2) 役割、(3) オブジェクト、(4) 数値、(5) 単位、(6) 対応、(7) 計算、の7つである。本章ではこの問題の診断に用いられている7つの要素を制約と定義し、TA を用いて学習者に制約を考えさせる。

1つでもこれらの制約に違反すると、提示された作問課題に対応する正しい文章題を作ることができない。例えば、(1) 文章構造は、「存在文二つ、関係文一つの数的一致するか」という制約となる。なお、ここでは詳細は省くが、存在文や関係文は、カードの種類を示している。存在文とは、「りんごが4個あります」「りんごが120gあります」といった存在そのものを示す文のカードであり、関係文とは、「りんごが一個あたり30gあります」といった、「1個」と「30g」の関係性など、関係性を表す文のカードである。例えば、これら3つのカードを組み合わせると、「 $30 \times 4 = 120$ 」になる文章題を作ることができる。

また、この7つの制約は、依存関係が一部存在する。例えば、「文章構造」の制約が誤っているときは必ず「対応」の制約も誤っている。しかし、「対応」の制約が誤って

いる場合に必ず「文章構造」の制約が誤っているとは限らない。モンサクンにおける学習活動や乗除算を対象とした作問学習の診断に用いられる 7 つの制約の詳細は文献 [前田 15, 山元 17, 平嶋 19] を参照されたい。

6.5.3 制約違反の診断

TAME では、制約を用いてカードの組み合わせを診断し、学習者が作った作問と照らし合わせて制約違反を調べる必要がある。そこで、乗除算を対象としたカードの組み合わせの診断方法について述べる。

9 つあるカードの全組み合わせと制約違反のパターンを調べる場合、すべての組み合わせは ${}_9P_3$ の計算から 504 通りとなる。カードが ad-hoc に記述にされると、制約違反の診断が困難になる。したがって、作問で使用するカードのテンプレートをあらかじめ決定する。例えば、「1 本あたり赤いリボンは 14cm です」というカードをテンプレートに変換すると、「1 (単位 1) あたり (オブジェクト 1) が (数字 1) (単位 2) です」という形になる。これらの各テンプレートには ID が設定されており、カードの組み合わせとその位置をシステム内で記録する^{*11}。

このような仕組みを用いることで、ad-hoc な記述に依存することなく、各スロットに入ったテンプレート、ID、スロットを記録することができる。また、テンプレート ID の組み合わせごとに起きる制約違反のパターンを数え上げることも可能になる。

6.5.4 実現可能性の評価

手続き

TAME の枠組みを用いることで、6.5.3 で示したような乗除算の診断が行えるかどうかはこれまでの説明からは自明ではない。本節では、6.4 の手法および 6.5.2, 6.5.3 で述べた制約・診断によって TAME が実現可能かどうかを実際のシステム構築を通した検証に

^{*11}組み合わせだけでなく位置も記録するのは、乗除算はカードの役割を考える必要があり、その役割を判断するためにスロットを用いているためである。例えば乗算では、スロットは「一つ分の数」、「いくつ分」、「全部の数」のように役割で分かれている。

よって評価する。具体的には、モンサクンの乗除算 [前田 15, 山元 17, 平嶋 19] における「一つ分の数」×「いくつ分」＝「全部の数」の関係を対象に、カードの組み合わせに対する制約を用いたカード生成、制約パターンの診断を行うことで、事例的に TAME の枠組みの概念的な実証を試みる。

準備

まず、モンサクンの乗除算において TAME を用いた TA を実現するためには、(A) 学習課題の実現に必要な情報と、(B) TAME の実現に必要な情報の 2 種類を用意する必要がある。

(A) には、カードの生成を行うための、6.5.3 で述べたようなカードのテンプレートを含めたカードに対する属性設定とテンプレートに当てはめる単語や数値などのデータセットが該当する。よって、表 6.1 に示すダミーを含めた 9 種類のテンプレートを用意した。表 6.1 のテンプレートでは、例えば「1 個あたりりんごは 150g です」という形式のカードについて、「1[Unit x] あたり [Object y] は [Number z] [Unit w] です」という抽象化を行っている。なお、表中の ID1-3 が正解の組み合わせに必要なテンプレートで、それ以外はダミーのテンプレートである。このテンプレートでは、2 種類の [Object]、3 種類の [Number]、3 種類の [Unit] の組み合わせで、ID1-3 の組み合わせ以外に正解が成立しないように構成されている。このようなテンプレートから具体的なカードを用意するために、表 6.2 に示すデータセットも用意した。表 6.2 のデータセットでは、2 種類の [Object]、3 種類の [Unit] に入る具体的なラベルと、3 種類の [Number] に当てはまる値をランダムに生成するための、取りうる値のレンジを用意している。どの [Unit] であるかに関わらず、この [Number] のレンジからランダムに値が生成される。このデータセットを用いることで、データセット ID ごとに表 6.1 のテンプレートをもとにした 9 枚のカード、すなわち、問題を用意することができる。さらにモンサクンの学習活動では、これらのカードを 3 種類のスロットに当てはめることで作問活動を行う。そのため、カードの属性に加えて、当てはめたスロットの位置が診断に関わる情報として加わる。

また、(B) については、上記の学習課題で規定された属性を用いた診断のための制約が該当する。そこで、上記の学習課題の構造化と乗除算の制約 (6.5.2) を踏まえて、表 6.3

に示す制約表現を用意した。表 6.3 のそれぞれの記法について説明する。前提として、制約は乗除算の作問における 3 つあるすべてのスロットがカードで満たされている際に、スロットに満たされたカードの組み合わせに対しての診断に用いられる。まず、 C_{1-3} はカードスロット 1-3 に挿入されたカードを示す。対象とする「一つ分の数」×「いくつ分」=「全部の数」の文章題においては、カードスロット 1 は「一つ分の数」、カードスロット 2 は「いくつ分」、カードスロット 3 は「全部の数」を示す。次に、 $\text{CountOf}(x)$ は x の数、 $\text{IsExist}(x)$ は x の存在の有無を示す。また、 $\text{Number}(x)$ や $\text{Type}(x)$ は、前者であれば x に示される数値、後者であれば x のカードの種類を指す。R5 と R7 は、R1 を包含した制約であるため、関係文が 1 つ、存在文が 2 つあることが保証される。したがって、存在文₁ や 存在文₂ はそのうちどちらかの 1 つ目と 2 つ目を指す。R6 では、特定の関係文 1 つの存在が確認されたときに、それ以外の 2 つのカード（関係文でも存在文でも良い）

表 6.1 用意したカードのテンプレート一覧

ID	種類	テンプレート文章
T1	関係文	1[Unit2] あたり [Object1] は [Number1][Unit1] です
T2	存在文	[Object1] は [Number2][Unit2] です
T3	存在文	[Object1] は [Number3][Unit1] です
T4	関係文	1[Unit2] あたり [Object2] は [Number1][Unit1] です
T5	存在文	[Object2] は [Number2][Unit2] です
T6	関係文	1[Unit2] あたり [Object2] は [Number1][Unit3] です
T7	関係文	1[Unit2] あたり [Object2] は [Number3][Unit1] です
T8	存在文	[Object1] は [Number3][Unit3] です
T9	関係文	1[Unit2] あたり [Object1] は [Number2][Unit1] です

表 6.2 用意したデータセットの一覧

ID	[Object1]	[Object2]	[Number] のレンジ	[Unit1]	[Unit2]	[Unit3]
D1	青いリボン	赤いリボン	10-20	cm	本	円
D2	青いリボン	赤いリボン	10-20	円	本	cm
D3	りんご	みかん	100-200	g	個	円
D4	りんご	みかん	100-200	円	個	g

表 6.3 用意した制約表現の一覧

ID	制約
R1	$\text{CountOf}(\text{関係文}) = 1 \wedge \text{CountOf}(\text{存在文}) = 2$
R2	$\text{Type}(C_1) = \text{関係文} \wedge \text{Type}(C_2) = \text{存在文} \wedge \text{Type}(C_3) = \text{存在文}$
R3	$\text{Object}(C_1) = \text{Object}(C_2) = \text{Object}(C_3)$
R4	$\text{IsExist}([\text{Number1}]) \wedge \text{IsExist}([\text{Number2}]) \wedge \text{IsExist}([\text{Number3}])$
R5	$R1 \wedge \{(\text{Unit}(\text{存在文}_1) = \text{Unit}(\text{関係文}_{\text{main}}) \wedge \text{Unit}(\text{存在文}_2) = \text{Unit}(\text{関係文}_{\text{sub}})) \vee ((\text{Unit}(\text{存在文}_1) = \text{Unit}(\text{関係文}_{\text{sub}}) \wedge \text{Unit}(\text{存在文}_2) = \text{Unit}(\text{関係文}_{\text{main}})))\}$
R6	$\text{IsExsist}(\text{関係文}) \wedge (\text{Number}(\text{その他の文}_{\text{upper-slot}}) < \text{Number}(\text{その他の文}_{\text{lower-slot}}))$
R7	$R1 \wedge \text{Number}(\text{関係文}) = [\text{Number1}] \wedge \{(\text{Number}(\text{存在文}_1) = [\text{Number2}] \wedge \text{Number}(\text{存在文}_2) = [\text{Number3}]) \vee (\text{Number}(\text{存在文}_1) = [\text{Number3}] \wedge \text{Number}(\text{存在文}_2) = [\text{Number2}])\}$

に記された数字が、上位のスロットにあるカードのほうが小さく、下位のスロットにあるカードのほうが大きいことを示す。

これらのテンプレート、データセット、制約を用いて、診断の検証を行う。

検証結果

本検証でシステムを用いてカードを生成した結果、すべてのデータセットにおいて不自然ではない形で9つのカードをテンプレートから生成できることを確認した。また、テンプレートから生成されたカード9つから、3つのカードの組み合わせを取り出す ${}_9P_3$ の504通りにおいて各制約ごとの充足・違反を診断できることを確認した。

7つの制約については、「充足する」もしくは「充足しない」の2値が含まれるので、 2^7 で128件の制約充足パターンが想定される。しかし、各制約は完全に独立ではなく、テンプレートの種類にも依存するため、今回用意したテンプレート（表6.1）と制約（表6.3）からは、重複しない（ユニークな）制約充足パターンの組み合わせが41件作れることがわかった。したがって、504通りのカードの組み合わせは、いずれもこの41件の制約充足パターンに分類されるといえる。なお、これらの41件の制約充足パターンと6.4.4に基づいて、どの制約充足パターンから開始した場合もR1-R7の各指摘に対応した制約充

足パターンを選定できることを確認した。

また、504通りのカードの組み合わせを、制約違反の数ごとに分類した。その結果、制約違反が0個の場合（つまり、正解のパターン）は1通り、1個の場合は9通り、2個の場合は27通り、3個の場合は39通り、4個の場合は36通り、5個の場合は81通り、6個の場合は173通り、7個の場合は138通りの組み合わせが存在することがわかった。さらに、それぞれ個々の制約にどれぐらいの組み合わせが違反しているのかを集計した。R1の制約に違反するパターンが324件、R2の制約に違反するパターンが420件、R3の制約に違反するパターンが468件、R4の制約に違反するパターンが342件、R5の制約に違反するパターンが444件、R6の制約に違反するパターンが327件、R7の制約に違反するパターンが408件確認された。

以上の検証から、表6.1に示すような学習課題において解答の操作に用いられる要素と、表6.3に示した解答の診断に用いる制約を用意することで、TAMEが実現できることがわかった。

6.5.5 TAMEに基づくTAの実装

モンサクンと6.5.4で説明した枠組みをベースに実際にシステム（以下、本システム）を開発した。本システムでは、6.5.4で説明した枠組みに基づいて、カードのテンプレート群から具体的なカードを生成する。また、学習者のカードの組み合わせを診断する際には、テンプレートレベルでの診断を行う。

本システムでは、問題演習フェーズと教えるフェーズの2つのフェーズで成り立っている。システムインタフェースのうち、問題演習フェーズの画面を図6.2に、教えるフェーズの画面を図6.3に示す。

問題演習フェーズは、従来のモンサクンと同等の作問演習である。図6.2のシステムインタフェースのうち、右部にあるカードのリストから左側の解答欄にドラッグアンドドロップすることで、解答を組み立てていく。解答を組み立て終わったら、右上部のボタンで解答（答え合わせ）をする。

教えるフェーズは、TAMEに基づいてTAに作問を教える演習を行う。図6.3のシステムインタフェースのうち、下部にある7つのボタンがそれぞれの制約に対応する制約違

反の指摘ボタンである。6.4.4 で説明した挙動制御に基づいて、TA は指摘された制約違反を修正するように解答を組み立て直す。よって、アルゴリズム 2 に基づき、学習者の指摘を制約レベルで必ず修正した制約充足パターン（すなわち誤り方）を満たす解答の中から組み立て直す解答が無作為に一つ選ばれる^{*12}。学習者が、TA の解答が正解だと思った時点で、問題演習フェーズと同様に解答ボタンを押し、TA の解答を診断してもらう。

問題演習フェーズで 1 問学習者が正答するたびに学習者が解いた問題と同じ問題の教えるフェーズに移行し、教えるフェーズで TA に正答させることができれば、再度問題演習フェーズで次の問題に移行する。

^{*12}なお、本ケーススタディで実装した TA においては、そのなかでも、カードそのものの入れ替えが最も少ない解答を選ぶ。これは、TA の修正の自然さを考え、暫定的に設定している。



図 6.2 システムインタフェース（問題演習フェーズ）

6.5.6 有効性の評価

手続き

TAME を用いた TA を学習支援システムに追加実装することが対象ドメインの制約を理解するのに有効かを評価するために評価実験を行った。具体的な仮説は、TAME を用いた TA を学習支援システムに追加実装することで、(H1) 対象ドメインの理解に有用である、また (H2) 対象ドメインの制約や誤り箇所の理解に有用である、とした。そこで、従来の学習支援システムに TAME を用いた TA を拡張実装したシステムと、従来の学習支援システムのみでの学習効果を比較するために、工学部大学生 15 名を対象に実験群 8 名と統制群 7 名に分けて実験を実施した。両群ともに事前テストを合計 20 分間で実施したのち、各群に分かれてシステム学習を 20 分間実施し、その後両群とも事後テストを合計 20 分間とアンケートを 5 分間で実施した。システム学習では、実験群には 6.5.5 で実装し

準備ができたなら、
右上のボタンで僕が答え合わせするよ！

答え合わせしてもらおう

問題：「 $12 \times 4 = 48$ 」になるお話を作ろう！

1つ分の数 1本あたり青いリボンは12円です

いくつ分 青いリボンは4本です

全部の数 青いリボンは48cmです

1本あたり青いリボンは4円です

1本あたり赤いリボンは48円です

青いリボンは48円です

1本あたり赤いリボンは12円です

赤いリボンは4本です

1本あたり赤いリボンは12cmです

今度はぼくが問題を解いてみたよ！
間違っているところがあれば教えてね！

文章構造の指摘 役割の指摘 オブジェクトの指摘 数字の指摘 単位の指摘 対応の指摘 計算の指摘

図 6.3 システムインタフェース（教えるフェーズ）

た本システムを、統制群には本システムの問題演習フェーズのみ（つまり、従来のモンサクンと同等のシステム）を用いて学習してもらった。また、システム学習においては、明示的に制約を示すか示さないかが学習効果の差にならないよう、7つの制約に対する説明資料を両群ともに配布した。

事前テストおよび事後テストでは各8問からなる5種類のテスト（計40問）を出題した。事前と事後の間で、実施したテストの種類は同じであり、テスト内の問題のみ変更している。以下に述べる各テストの時間配分は、工学部大学生を対象とした予備実験を通して設定したものである。また、以下に述べる順番で実験中もテストを実施している。

一つ目の誤り箇所指摘テストでは、4分間で、各問で3枚のカードのうち1枚が誤った状態で提示されるので、誤っている1枚を見つけ出し、そのカードの文中で誤っている要素を抜き出すことが求められる。このテストでは、間違い箇所について指摘させることで、学習者が制約違反を犯している文やその要素を理解できるかを問うことを目的としている。

二つ目の役割指摘テストでは、2分間で、各問で正答を組み立てることができる3枚のカードが提示されるので、システムインタフェースの左部にあったように、正しい順番で役割のロットに入れて組み立てることが求められる。このテストでは、制約の中でも最も基礎となる文の役割について理解しているかを問うことを目的としている。

三つ目のカード入れ替えテストでは、4分間で、各問で既に組み立てられた3枚のカードの組み合わせ（誤答）が提示されるので、間違っているカードを選択し、別途提示された選択肢から正しいカードを選んで入れ替えることが求められる。このテストでは、学習

表 6.4 各テストの結果

群	誤り箇所指摘		役割指摘		カード入れ替え	
	事前	事後	事前	事後	事前	事後
実験	2.13 (1.27)	6.50 (1.66)	4.38 (1.22)	6.13 (1.69)	2.63 (0.99)	2.88 (1.76)
統制	2.86 (1.46)	4.00 (1.60)	4.43 (1.68)	6.29 (1.28)	1.71 (1.03)	2.57 (1.92)
群	作問		制約指摘 (正)		制約指摘 (誤)	
	事前	事後	事前	事後	事前	事後
実験	4.25 (2.05)	5.25 (2.38)	10.75 (3.15)	12.25 (4.58)	-7.00 (3.97)	-3.13 (2.20)
統制	2.86 (1.36)	4.71 (1.58)	11.43 (3.45)	14.00 (4.72)	-7.14 (3.83)	-6.86 (3.80)

注: () の外の値は平均, 内の値は標準偏差を示す。

者が選択肢の中から適切なカードを選び、誤りを修正する能力を問うことを目的としている。

四つ目の作問テストでは、5分間で、システム中の学習活動と同様に、各問で提示されたカードの選択肢から、問題文に当てはまるようにカードを組み立てることが求められる。このテストでは、全般的な作問能力を問うことを目的としている。

五つ目の制約指摘テストでは、5分間で、3つ目のテストと同様に各問にて誤答が提示されるので、間違っている制約をすべて指摘することが求められる。このテストでは、各制約に対する具体的な理解度を問うことを目的としている。

なお、これまでのモンサクンの研究から、加減算において大学生であっても小学生と同様の制約の誤り傾向を持つことが確認されており、また、モンサクンが大学生にも学習効果を発揮することを示している [Hasanah 19]。言い換えれば、制約を学習することの困難性を示しているとも考えられる。よって一般的に加減算より難易度の高い乗除算においても、同様のことが考えられるため、本実験には一定の妥当性があると考えている。

結果

制約指摘テストを除いた4つのテストは、1問あたり完答で1点として採点した。制約指摘テストは、問題毎に指摘すべき制約の数が異なるため、指摘ごとに1点とした。また、制約指摘テストでは、すべての制約を指摘すれば必ず正しい指摘を網羅できてしまう。そのため、正しく指摘できた数だけでなく、違反していない制約を指摘した数も分けて評価することとした。誤った制約指摘では、誤った指摘ごとに-1点として、別途採点した。各テストの結果を表6.4に示す。

各テストの点数を対象に、群（実験・統制）を実験参加者間要因、テストのタイミング（事前・事後）を実験参加者内要因とした 2×2 の分散分析を実施したので、以降で述べていく。誤り箇所指摘テストでは、群内に有意差（ $p < .05$ ）がみられた。また、交互作用が確認されたため、単純主効果の検定を実施した。その結果、実験群における事前と事後の間、統制群における事前と事後の間、事後における実験群と統制群の間に有意差（ $p < .05$ ）がみられた。役割指摘テストでは、群内にのみ有意差がみられた（ $p < .05$ ）。カード入れ替えテストでは、有意差がみられなかった。作問テストでは、群内にのみ有意

差がみられた ($p < .05$). 制約指摘テストでは、正指摘数においては、有意差がみられなかった。一方で、誤指摘数については、交互作用が確認されたため、単純主効果の検定を実施した。その結果、実験群における事前と事後の間と、事後における実験群と統制群の間に有意差がみられた ($p < .05$).

アンケート結果

アンケートでは5「とてもそう思う」～1「とてもそう思わない」の5段階で回答してもらった。

次の2つは、作問活動を行うことで制約理解につながるかを問う質問である。「作問活動はカードの組み合わせに必要な制約を意識することにつながるか」という質問では実験群が4.13、統制群が4.14とどちらも高い結果となった。「作問活動はカードの組み合わせに必要な制約の理解の向上につながるか」という質問では実験群が3.88、統制群が3.71とこちらも同様に高い結果となった。

また、次の2つは実験群のみに質問した項目であり、教える活動を行うことで制約理解につながるかを問う質問である。「教える活動はカードの組み合わせに必要な制約を意識することにつながるか」という質問では4.00、「教える活動はカードの組み合わせに必要な制約の理解の向上につながるか」という質問でも同様に4.00と、高い評価を得ている。

考察

各テストの結果について、考察する。議論の都合上、実験で実施した順番と異なる順番で考察を述べる。

まず、誤り箇所指摘テストの結果から、教えるフェーズの活動が制約違反をしている文やその要素についての理解に効果があることが示唆された。また、制約指摘テストにおける誤指摘数の結果においても、実験群の誤指摘が減少していることから、教えるフェーズの活動が制約指摘の正確性を向上させていると考えられる。よって、教えるフェーズの活動によって制約理解が促されたことが示唆された。

役割指摘テストと作問テストの結果から、作問学習に対して、統制群が行った従来の活動だけでも一定の効果を得られることが示唆された。同時に、実験群は統制群に比べ問題演習フェーズの時間は少なく、その分教えるフェーズの時間が入っているが、問題演習

フェーズ単体と差異のない結果が得られたことから、教える活動を通して文の役割についての理解や全般的な作問能力に寄与することが示唆された。

カード入れ替えテストの結果については、本来実験群と統制群どちらの学習においても一定の点数向上がみられることを期待していたが、そういった向上はみられなかった。こちらは、今後個別にテストの内容や状況を詳細に検討する必要がある。

まとめると、TAMEによって実現される教える活動は、従来の作問学習支援同様に作問学習の理解に有用であることと、制約や誤り箇所理解においては、従来の作問学習支援を上回る理解が得られることが示唆された。よって、H1は支持されなかったが、H2は支持される結果となった。さらに、アンケートの結果から、学習者が作問活動を学習に有意義であると評価しており、教える活動についても同様に高く評価していることが分かった。したがって、TAMEで実現された学習活動を既存システムに追加することに一定の有用性があることが本ケーススタディを通して確認された。

6.5.7 全体的な考察

6.5.6における実験群と統制群の間におけるシステムの差異は、教える活動を有するか否かであり、誤り箇所指摘テストや制約指摘テストにおいて差が見られたことや、アンケートで教える活動が制約理解につながったという評価結果は、TAの振舞いが制約の理解に貢献したことを示唆する。つまり、本章で実現したTAに制約を教える活動が、制約の理解を促進したこととなり、これは本制御機構により実現されるTAの挙動が、学習者にとって理解可能であったことを意味する。このことにより、本章では再帰的フィードバックが実現できた可能性が示唆され、課題(1)(2)の解決が見られたのではないかと考える。

さらに、上記の解決について、6.5.4における実現可能性の評価において、既存の学習支援システムが備えている情報を用いてTAMEによるTAの制御の実現を実証的に確認できたといえる。

以上より、本章が提唱したフレームワークでTAを制御しうること、および制御されたTAの振舞いは、制約の理解に有益であり、再帰的フィードバックを実現できていたと思われることが示され、本章の有効性が示されたといえる。

しかしながら、本検証は TAME を用いた再帰的フィードバックの制御の有効性を確認するに留まっており、学習者が TA を共同学習者として認知したか否かについては検証できていない。Okita ら [Okita 07] は、TA をコンピュータとして認識するか、人間として認識するかによって学習効果が異なることを確認しており、TA の外見や話し方、感情制御などの要素が社会的スキーマにおいては重要であると考えられるため、それらの追究については今後の課題である。

6.6 本章のまとめ

本章では、正解や学習者の解答を充足すべき制約のパターンとして抽象化し、充足すべき制約のパターンとそれに基づく挙動の生成を行う挙動制御手法として、TAME を提案した。また、ケーススタディとして、学習支援システム「モンサクン」[前田 15, 山元 17, 平嶋 19] を対象に TAME を導入する概念的な実証と、それに基づいた TA を伴う学習支援システムの開発を行った。システムを用いた評価実験の結果から、TAME に基づく TA は、学習における一定の有効性が示唆された。

一方、現状の TAME のみでは、TA を実装するための全機能をカバーできない。Biswas ら [Biswas 05] は、TA を設計するためには (1) TA が情報を知覚するための監視機能、(2) 事前知識および情報 (入力) を保管する記憶機能と、記憶機能に基づく意思決定機能、(3) 対話や行動を制御する実行機能、の 3 点が必要だとしている。TAME では、このうち (1) と (2) を実現しているが (3) は実現しておらず、ケーススタディでも簡易的な実装に限定しているため、今後の課題である。

また、本章では、TA の挙動制御手法を中心に本章の価値をまとめることに努めたが、ケーススタディで取り扱った作問学習支援分野においても一定の価値を持つと考えている。例えば、モンサクンにおいても学習課題において満たすべき制約を学習者が理解することが重要視されている [平嶋 19]。したがって、学習者がモンサクンにおいてどのように制約を理解しているのかといったことを調査するために、学習プロセスの分析が盛んに行われてきた (こういった調査からは、学習者が理解しづらい制約や、頻繁に起こす制約違反のパターンが明らかになってきている) [Supianto 17b, Hasanah 17, Hasanah 19]。本章で開発された TAME に基づく TA は、これまで作問学習支援において重要視されて

きた制約の理解にアプローチするための新規な手段と捉えられる。本章では提案手法の実証を主たる目的としたため大学生を対象とした実験に留まったが、本アプローチの有効性を精緻に評価するためには、実際に乗除を学ぶ小学生を対象とした評価が課題である。TAMEの教えるフェーズのように、小学生に制約を直接扱わせることは、普段暗黙的に考えていることを明示的に思考する必要がある、一定の難易度が懸念される。しかし、すでに小学生に向けた制約の理解支援の取り組み [鳴川 21] も試験的になされており、小学生にとって理解しやすい表現を用いることによって、一定の学びが得られるのではないかと期待している。

第7章 ロボットの独自表現である目の発色を用いた学習者の感情制御

7.1 本論文における本章の位置付け

6章では、情報構造指向アプローチに基づく領域知識モデルに対して汎用的な教授モデルを提案し、その実現可能性と有効性を示した。これにより、情報構造指向アプローチに基づく領域知識モデルが再利用しうることで、どのような教授モデルおよび領域知識モデルの組み合わせであれば領域知識モデルを再利用できるかについての知見を示唆した。

一方で、6章で示した教授モデルは、学習者の知識状態のみを支援対象に含んでおり、学習者の感情状態は対象範囲に含まれず、考慮できていない。そこで本章では、6章で示した教授モデルと独立し、かつ併用できる感情的支援の方法について模索する。具体的には、ITSのインタフェースにロボットを用いることで、ロボットの目の色と発話の組み合わせにより学習者の感情を制御しうることを示す。これにより、情報構造指向アプローチに基づく領域知識モデルを汎用的に再利用できる知識的支援および感情的支援を含む教授モデルを実現しうることを示唆する。

7.2 ロボットを用いた学習支援としての本章の動機

科学技術の発展に伴い、人間の教育者・講師をロボットに置き換える試みが行われている。Kamideら [Kamide 14] は、人型ロボットがプレゼンテーションを行う際の非言語行動として、画面の要点を強調する行動や聴衆の注意を引きつける行動が重要であると報告している。また、Ishinoらは、自身の非言語行動を制御できるロボットは、学習者が注目したい部分を視線やジェスチャー、パラ言語で強調し、その行動によって学習者の興味を促進できることを示唆した [Ishino 18, Ishino 22]。しかし、これまでロボット特有の非言語行動を講義に活用する試みはなかった。

色は人の心理に影響を与え (e.g., [Gao 06, Elliot 07])、人は色によって異なる感情を抱く。人間の先生は授業中、自然に表情が変わる一方、ロボット講師は目の色を変えるなど、ユニークな行動をとることができる。もし、授業中に先生の目が赤くなったら、学習

者はどのような感情を抱くのか？このような振舞いがどのような感情を呼び起こすかを調べることで、ロボットによる教育への幅広い応用が期待される。

学習者の精神状態に応じて適切に対応することの重要性が、研究において強調されている [Dennis 16]。このように、学習時には教材だけでなく、学習者に提示される感情も考慮する必要がある。例えば、教室での授業中、教師が笑っているか、怒っているか、あるいは悲しんでいるかは、学習者に喚起される感情に強く影響することがある。学習者は、怒っている先生を前にするとストレスを感じ、笑顔の先生から学ぶとリラックスする。このように、学習・教育・学力に直接関係する感情を、Pekrun らは「Academic Emotions」と呼んで [Pekrun 02]。そこで、外的要因を用いて学習者の Academic Emotions を向上させる方法を検討することが重要であると考えた。

本章では、ロボットの目の発光色によって喚起される Academic Emotions を調べるために、(A) 学習者の Academic Emotions を操作できるか、(B) 学習者に適応的に Academic Emotions を生み出せるか、(C) 実際の講義と同様の状況で、Academic Emotions を操作できるか、を仮説として検証した。将来的には、ロボット講師が講師に代わるだけでなく、理想的なロボット生徒（理想的な質問をするなど、授業を推進する行動をする生徒）を教室に加えることも可能である。そこで、ロボットの役割を講師と学習者に分け、色の効果を調査した。仮説 (A), (B), (C) の検証は、それぞれ 7.4 節, 7.5 節, 7.6 節で述べる。

7.3 関連研究

7.3.1 教育におけるロボット

近年、教育用途でロボットが注目されている。Deublein ら [Deublein 18] は、ロボットの動機づけ行動により、教育現場での学習成果を向上させることを示唆した。また、Ishino ら [Ishino 18, Ishino 22] は、プレゼンテーション講義において、ロボットが非言語的行動を制御することで学習者の興味を促進できることを示唆した。

加えて、Jimenez ら [Jimenez 15] は、ロボットの感情を表現することで、学習者との協働学習を促進することを示唆している。また、Jimenez ら [Jimenez 18] は、ロボットの

感情表現が学習者との協働学習においてどのように作用するかを検証している。このように、教育における人間とロボットの相互作用の中で感情が発生することも分かっている。しかし、学習者に与えられる感情については明らかにされていない。

7.3.2 Intelligent Tutoring System

近年、学習者の知識状態だけでなく、精神状態も支援しようとする ITS (Intelligent Tutoring System) の研究が盛んになっている。例えば、ITS における Intelligent Mentoring System [Kojima 12, Muramatsu 16] は、学習者から解答履歴やマウス操作などの粒度の異なる情報を取得し、知識状態や心理状態を支援する試みである。AutoTutor [D’Mello 07] は、ITS が学習者の感情を読み取る、より実用的な方法の一つを示したものである。しかし、これらの研究は学習者の感情を読み取るためのものであり、ITS の出力から学習者の感情を制御するためのものではない。

7.4 検証 (A) : 学習者の Academic Emotions を操作できるか？

7.4.1 概要

Pepper (Softbank Robotics) の色彩機能が学習者の学習活動に与える Academic Emotions への影響を検証するための実験を行った。参加者は、大学院生と学部生の計 10 名である。

7.4.2 目的

本実験では、以下の仮説の検証を目的とした。(A-1) ロボットの講義行動の一部として色情報を提示することにより、学習者の Academic Emotions に何らかの影響を与える。(A-2) 色情報を提示することにより、学習者の Academic Emotions が操作できる。

7.4.3 刺激

Pepper の講義行動として、4つの発話パターンを用意した。講師役と学習者役でそれぞれ2つの発話パターンを用意した。

- 重要: 今から説明するポイントは重要です (講師役)
- 注意: 授業に関係ない話はやめてください (講師役)
- 困惑: 困った, わからない (学習者役)
- 理解: なるほど, そうか (学習者役)

今回の色表現では, Pepper の目の LED の色だけを変え, 他の部分の色や姿勢は変えなかった (図7.1)。また, 色表現のみを検証するため, ジェスチャーは行わなかった。さらに, 色によって喚起される感情は主観的に解釈されるため, 色と感情の対応関係を以下のように示すことで, 解釈を限定した。

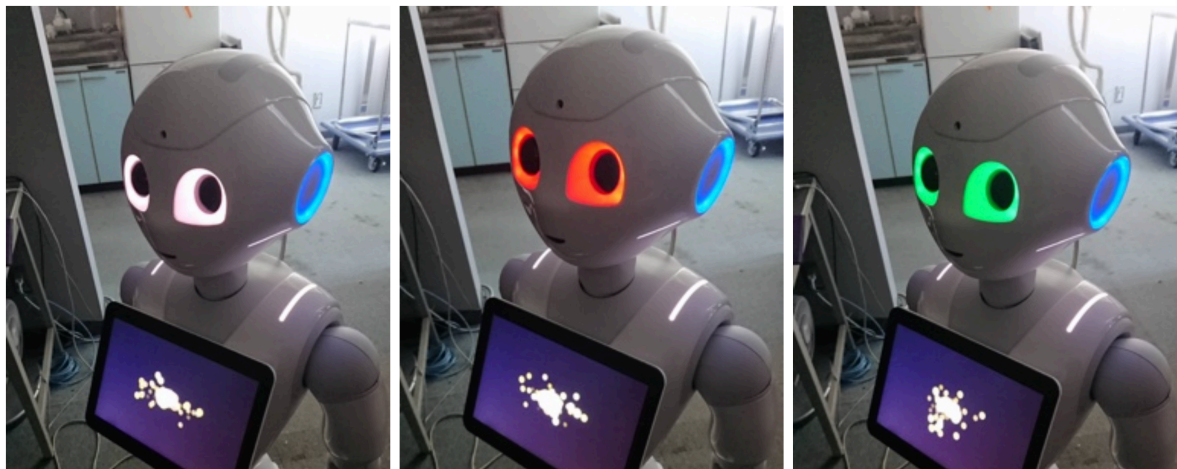


図 7.1 Pepper の目のそれぞれ異なる発色 (左から, 白, 赤, 緑)

- 赤: 怒り, もしくは強い感情
- 緑: 喜び, もしくは弱い感情
- 白: 無感情

実験では, 4つの発話パターンと3つの色の12通りの組み合わせが刺激として用いら

れた。

7.4.4 質問紙

心理学では、学習・教育・学業に関連する感情を Academic Emotions と呼んでいる。Academic Emotions に含まれる感情は、楽しさ (enjoyment)、退屈 (boredom)、怒り (anger)、希望 (hope)、不安 (anxiety)、絶望 (hopelessness)、誇り (pride)、安堵 (relief) である [Pekrun 02, Pekrun 11]。これらの Academic Emotions を評価項目とした 7 段階のリッカート尺度による質問紙を作成した。

7.4.5 手続き

実験では、Pepper の全発話パターンを白、赤、緑の順で与えた。各発話パターンは、「このロボットと一緒に講義を受けるとどう感じますか？」という質問でアンケート評価を行った。その手順は以下の通りである (図 7.2)。

- 1) 参加者は Pepper の前に座る。
- 2) 参加者は Pepper の講義行動を評価する。
 - a) Pepper が目 (白) を光らせながら、[重要] の発言をする。
 - b) 参加者は、アンケートで講義の様子を評価する。
 - c) [] の発話パターンを注意、困惑、理解に変えながら、a) と b) を繰り返す。
 - d) すべての発話パターンが評価されたら、() の色を変え、a)~c) を繰り返す。

7.4.6 結果

結果を図 7.3、図 7.4、図 7.5、図 7.6 に示す。なお、Friedman の検定により有意差 ($p < 0.05$) を確認した後、Ryan の方法による多重比較を行った。図中の有意記号は Ryan の方法で算出した有意差 ($p < 0.05$) を示し、エラーバーは標準誤差を示す。

図 7.3 は、「重要」の発言パターンについての結果である。白と赤、白と緑では、退屈、

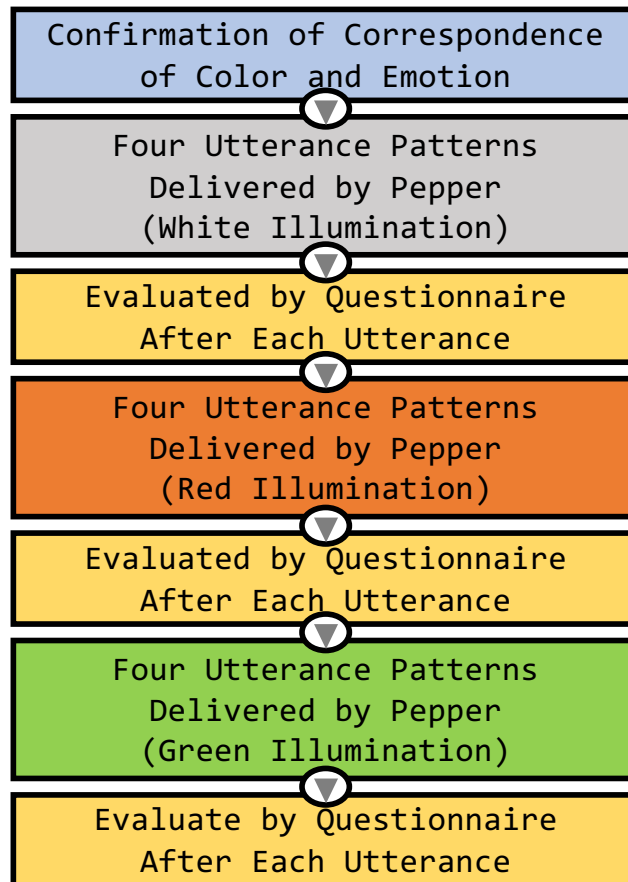


図 7.2 検証 (A) の手続き

誇り、希望、安堵に有意差が見られた。白は退屈を増強し、誇り、楽しさ、希望、安堵を抑制した。逆に、赤と緑では、退屈が抑制され、誇り、楽しさ、希望、安堵が増強された。また、白と緑では楽しさに有意差があり、白は楽しさを抑制し、緑は楽しさを増進していることが示唆された。

図 7.4 は、「注意」の発言パターンの結果である。白と赤の間には、希望において有意な差がみられた。白は希望を抑制し、赤は希望を増進させた。警告の文脈そのものが学習者個人に依存するためか、数値そのものには多くの違いが見られたが、それ以外には有意な差は見られなかった。

図 7.5 は、「困惑」の発言パターンの結果である。白と赤では、希望と安堵に有意な差が見られた。白は希望と安堵を抑制し、赤は希望と安堵を増強した。また、白と緑の間には、楽しさにおいて有意差がみられた。白は楽しさを抑制し、緑は楽しさを増強した。

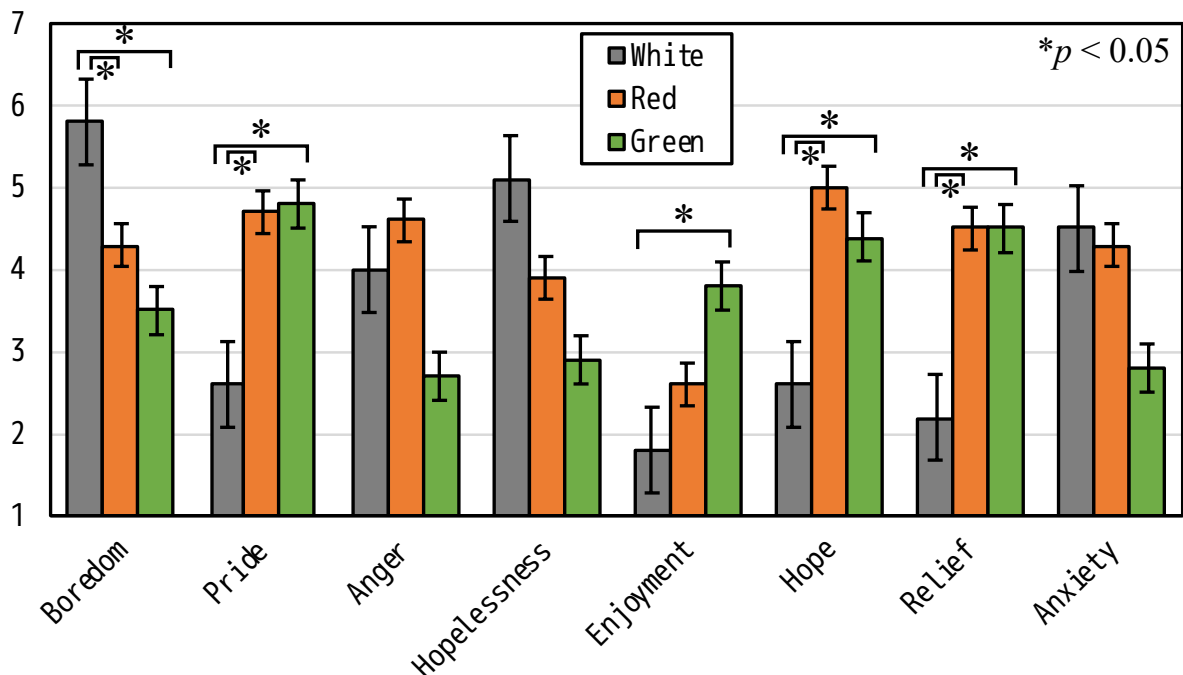


図 7.3 発話パターン「重要」における結果

図 7.6 は、「理解」の発話パターンについての結果である。白と緑、赤と緑では、誇り、絶望、楽しさ、不安で有意差がみられた。緑は、誇りと楽しさを高め、絶望と不安を抑制している。逆に、白と赤では、誇りと楽しさが抑制され、絶望と不安が増強された。また、白と緑の間には、退屈、希望、安堵において有意差があった。白は希望と安堵を抑制し、緑は希望と安堵を増強し、白は退屈を増強し、緑は退屈を抑制した。また、赤と緑の間には、怒りに対して有意差があった。赤は怒りを増強し、緑は怒りを抑制した。これらの結果から、緑は「理解」の発言パターンに良い影響を与えることが示唆された。

7.4.7 評価

これらの結果から、発話と色の組み合わせによる効果が異なること、学習者の Academic Emotions は色に依存し、色を変えることで操作可能であることがわかった。すなわち、仮説 (A-1), (A-2) が検証された。

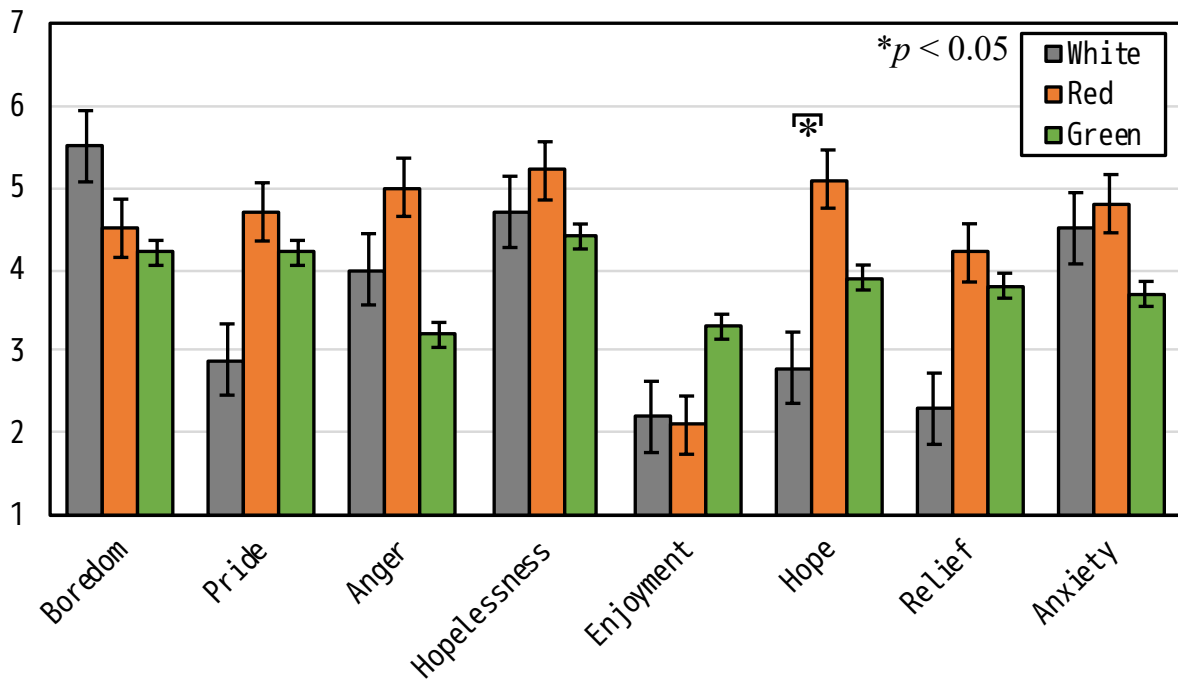


図 7.4 発話パターン「注意」における結果

7.5 検証 (B) : 学習者に適応的に Academic Emotions を生み出せるか？

7.4 節では、色を変えることで学習者の学業に対する感情を変化させ、操作できることを明らかにした。しかし、すべての学習者プロファイルを同じに扱うことはできず、学習者個人の特性に対応した色彩表現が望まれる。そこで、ここでは、教育心理学に関するアンケートを用いて、学習者の特徴を調査・分析した。

7.5.1 個別性を調査するための質問紙

西村らは、自己決定理論における内的調整スタイル、同一化調整スタイル、取り入れ調整スタイル、外的調整スタイルを評価する自己調整質問紙の日本語短縮版 [西村 11, Nishimura 17] を開発した。各調整スタイルは 5 つの項目で測定された。項目は 4 段階評価 (1: 強く反対-4: 強く賛成) であった。西村ら [西村 11] は、オリジナルの自己調整質

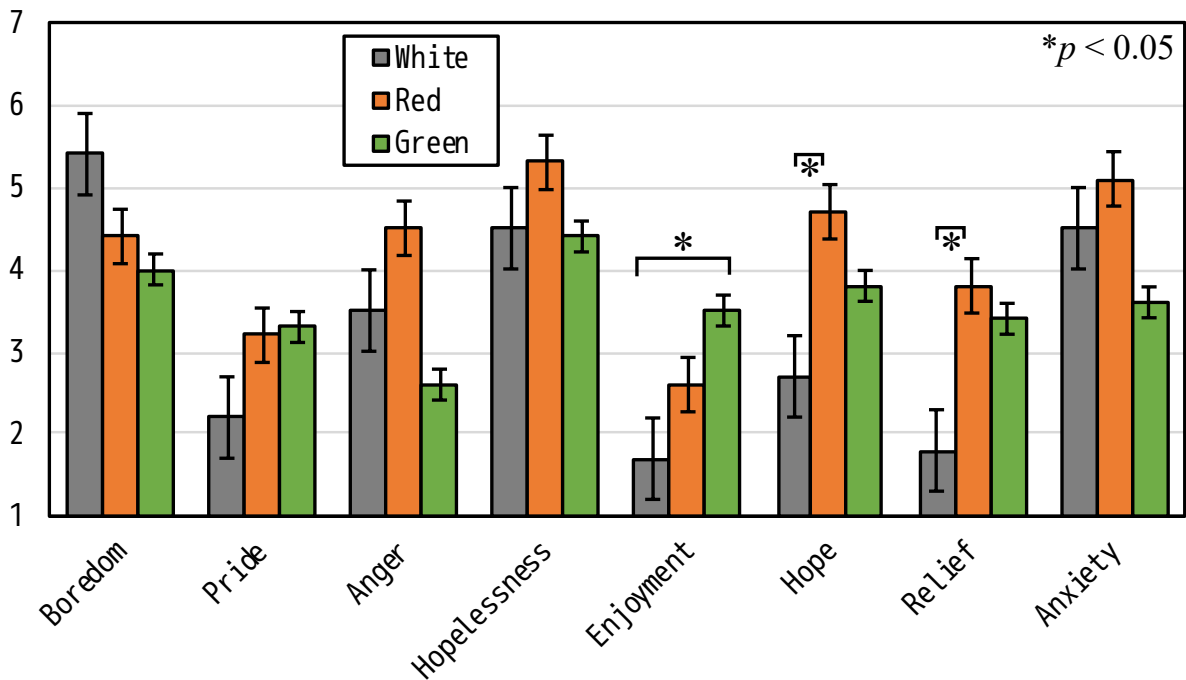


図 7.5 発話パターン「困惑」における結果

問紙 [Ryan 89] との相関分析により、この尺度の妥当性を報告している。

それぞれの調整スタイルの特徴は以下のとおりである。

- 内的調整：学習そのものに目的がある。例えば、学習そのものが面白い、望ましいなど、従来の内発的動機付けに相当する。
- 同一化調整：学習が自分の将来を形成することを認識するなど、活動を行うことの価値を認識し、それを自分のものとして受け入れることを表す動機付けである。
- 取り入れ調整：他者との比較による自己拡大や自己価値の維持など、活動の価値が否定的であるが部分的に内面化された動機付け。
- 外的調整：報酬獲得や罰の回避に影響され、従来の外的動機付けに相当する。

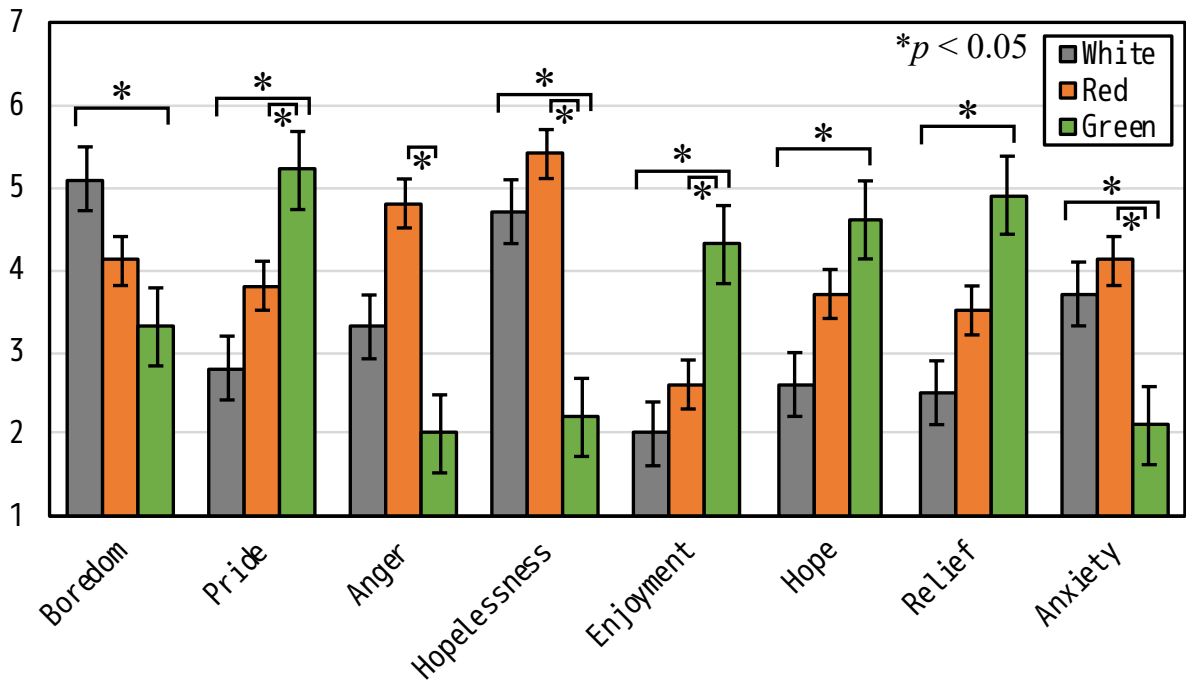


図 7.6 発話パターン「理解」における結果

7.5.2 方法

学習者の個性を調べるために、7.4節の参加者に自己調整質問紙 [西村 11] に回答してもらい、4つの調整スタイルごとに値を算出した。調整スタイルと8つの Academic Emotions の値を相関させることで、学習者ごとの個別プロファイルを取得し、学習者に適応的なインタラクションを事前に作成することが可能となった。

7.5.3 結果

表 7.1 は、各発話パターンに対する相関の結果を示したものである。各結果は非相関の検定を行い、有意差を示した (+ : $p < 0.10$ (斜体), * $p < 0.05$ (太字))。表 7.1 の結果から、発話パターン、色、学習感情、規則の組み合わせにより、有意差が生じる箇所があることがわかった。緑と有意な正負の相関を持つ調整スタイルがあった (表 7.2)。また、白と緑との誇りについても同様の結果が得られた (表 7.3)。したがって、調整スタイルと

Academic Emotions は関連しているといえる。

7.5.4 評価

調整スタイルによって複数の Academic Emotions を同時に呼び起こすことができる (e.g., 赤は怒りや楽しさと相関している) (表 7.3)。そのため、単一の Academic Emotion を単独のアクションで想起させることは困難であった。そこで、Academic Emotions を

表 7.1 発話パターン「重要」における調整スタイルと Academic Emotions の相関

色	調整 スタイル	退屈	怒り	絶望	不安	誇り	楽しさ	希望	安堵
白	内的	0.43	0.13	-0.09	0.30	0.75*	0.18	0.53	0.31
	同一化	-0.06	-0.02	0.74*	0.17	0.04	-0.48	-0.20	-0.16
	取り入れ	-0.11	-0.35	0.88*	-0.25	-0.08	-0.31	-0.38	-0.25
	外的	0.24	0.33	0.25	0.16	-0.46	0.01	-0.35	-0.14
赤	内的	-0.11	-0.32	0.35	-0.41	-0.25	0.16	<i>-0.62+</i>	-0.40
	同一化	0.17	0.51	0.17	0.17	-0.10	-0.64*	-0.06	-0.12
	取り入れ	0.27	0.43	-0.50	0.28	0.48	-0.54	<i>0.62+</i>	0.37
	外的	0.07	0.32	-0.06	0.10	0.08	-0.24	0.54	0.33
緑	内的	0.26	-0.15	-0.38	0.51	0.04	0.47	0.32	0.16
	同一化	-0.10	-0.20	-0.03	-0.08	0.23	-0.20	0.10	0.25
	取り入れ	-0.04	-0.06	0.15	-0.42	0.22	-0.28	-0.11	0.09
	外的	-0.43	-0.28	-0.05	-0.22	0.24	-0.38	0.23	0.13

表 7.2 発話パターン「注意」における調整スタイルと Academic Emotions の相関

色	調整 スタイル	退屈	怒り	絶望	不安	誇り	楽しさ	希望	安堵
白	内的	0.16	-0.26	-0.18	0.04	0.10	-0.38	0.34	-0.47
	同一化	0.30	0.33	0.43	0.08	-0.28	-0.10	0.12	-0.44
	取り入れ	0.13	0.18	0.42	-0.04	-0.01	0.45	0.34	-0.17
	外的	0.25	0.52	<i>0.57+</i>	0.22	<i>-0.57+</i>	-0.17	-0.43	-0.25
赤	内的	-0.03	-0.27	0.03	-0.19	-0.39	0.26	0.25	<i>-0.61+</i>
	同一化	0.27	-0.33	0.45	0.18	-0.19	-0.43	-0.15	-0.15
	取り入れ	0.75*	0.15	0.38	0.24	0.40	-0.40	-0.12	0.46
	外的	0.10	-0.19	0.07	-0.10	0.27	-0.34	0.21	0.36
緑	内的	0.14	0.31	-0.38	0.14	-0.03	0.77*	0.39	0.34
	同一化	-0.32	-0.02	-0.22	-0.34	<i>0.58+</i>	-0.06	<i>0.61+</i>	0.42
	取り入れ	-0.08	-0.07	0.23	-0.34	0.50	-0.42	0.17	0.05
	外的	-0.20	-0.51	0.19	-0.25	0.43	-0.68*	0.04	0.15

ポジティブ感情とネガティブ感情に分割して単純化し、どのような相互作用がポジティブ感情とネガティブ感情を引き起こすかを評価式で検討した。

$$E_{c,i,s} = \sum_{n=1}^4 (-1)r_{c,i,s,n} + \sum_{p=1}^4 r_{c,i,s,p} \quad (7.5.1)$$

学習者の感情を総合的に評価するために構築した評価式を式(7.5.1)に示す。色*c*の発

表 7.3 発話パターン「困惑」における調整スタイルと Academic Emotions の相関

色	調整スタイル	退屈	怒り	絶望	不安	誇り	楽しさ	希望	安堵
白	内的	0.39	0.52	0.45	0.48	0.72*	-0.17	0.46	-0.18
	同一化	-0.06	-0.32	0.37	0.31	-0.07	-0.30	-0.25	-0.2
	取り入れ	-0.24	-0.45	0.03	-0.04	-0.50	<i>-0.58+</i>	-0.44	-0.33
	外的	0.11	-0.66*	0.19	0.16	-0.68*	-0.23	-0.36	-0.11
赤	内的	-0.31	-0.37	<i>-0.58+</i>	-0.40	<i>-0.57+</i>	0.09	-0.68*	-0.39
	同一化	0.44	0.70*	0.10	0.37	-0.28	-0.70*	-0.12	0.35
	取り入れ	0.42	0.72*	0.15	0.44	-0.04	-0.50	0.11	0.54
	外的	0.49	0.47	<i>0.58+</i>	0.45	-0.17	-0.25	0.43	0.46
緑	内的	-0.03	0.39	-0.53	0.32	0.80*	0.52	0.47	0.75*
	同一化	0.08	-0.17	0.49	0.27	-0.23	-0.43	0.08	0.15
	取り入れ	-0.26	-0.54	0.54	-0.35	-0.52	-0.52	-0.04	-0.03
	外的	0.48	-0.40	0.44	0	-0.68*	-0.47	-0.40	-0.33

表 7.4 発話パターン「理解」における調整スタイルと Academic Emotions の相関

色	調整スタイル	退屈	怒り	絶望	不安	誇り	楽しさ	希望	安堵
白	内的	0.46	<i>0.59+</i>	0.47	0.35	0.04	-0.20	0.03	-0.03
	同一化	0.22	-0.52	0.14	0.11	-0.37	-0.79*	0.44	-0.09
	取り入れ	-0.01	-0.41	-0.21	0.11	-0.34	<i>-0.63+</i>	0.13	-0.17
	外的	0.15	-0.35	0.13	0.07	-0.22	0.12	0.47	0.12
赤	内的	-0.24	<i>-0.61+</i>	-0.45	0.12	-0.13	0.80*	<i>0.58+</i>	0.31
	同一化	0.22	-0.02	0.05	0.05	0.41	-0.32	0.23	0.28
	取り入れ	0.41	0.23	0.14	-0.35	0.85*	-0.38	-0.06	0.18
	外的	0.29	0.18	0.33	-0.04	0.10	-0.49	0.17	0.36
緑	内的	-0.47	-0.47	0	0.04	0.17	0.66*	0.48	<i>0.58+</i>
	同一化	0.39	0.17	0.17	0.43	-0.03	-0.35	0.02	0.06
	取り入れ	0.23	0.28	-0.37	-0.19	0.06	-0.39	-0.11	-0.14
	外的	0.51	-0.22	-0.44	-0.27	<i>0.55+</i>	-0.46	0.14	0.04

話 i に対する調整スタイル s の $E_{c,i,s}$ の評価は、正の感情（誇り、楽しさ、希望、安堵）（ $p = 1$ to 4）に対して 1、負の感情（退屈、怒り、絶望、不安）（ $n = 1$ to 4）に対して -1 とした。各 Academic Emotions と各調整スタイルの相関（ $r_{c,i,s,n}$ または $r_{c,i,s,p}$ ）を掛け合わせ、合計値を算出した。

個人適応性がない場合、相関は低くなり式 (7.5.1) の値も 0 に近づく。また、ある調整スタイルが、正の項目には正の相関が高く、負の項目には負の相関が高い場合、式 (7.5.1) は正の値をとり、その逆の場合は負の値をとる。従って、この値が大きいと自己調整スタイルとの相性が良いことを示し、小さいと相性が悪いことを示す。つまり、学習者にポジティブな感情を持たせたい場合は、発言と色の組み合わせによって、式 (7.5.1) が正の値を取るようなインタラクションを行うことが効果的となる。逆に、ネガティブな感情を持たせたい場合は、式 (7.5.1) がネガティブな値を取るようなインタラクションを行うことが効果的といえる。

表 7.5 には、式 (7.5.1) の結果を示す。1 以上の絶対値は斜体で、2 以上の絶対値は太字で表示している。

その結果、ロボット講師が重要な情報を提示していることを伝えるには、全体的に緑が効果的であり、特に外的調整スタイルの場合は、緑が効果的であることがわかった。また、ロボット講師が学習者に注意を発している場合でも、全体としては緑が効果的だが、

表 7.5 評価式に基づく調整スタイルと発話パターンの相性

色	調整 スタイル	重要	注意	困惑	理解
白	内的	0.99	-0.16	<i>-1.02</i>	-2.02
	同一化	<i>-1.64</i>	<i>-1.84</i>	<i>-1.11</i>	-0.77
	取り入れ	<i>-1.20</i>	-0.08	<i>-1.15</i>	-0.48
	外的	<i>-1.92</i>	-2.97	<i>-1.19</i>	0.50
赤	内的	-0.61	-0.03	0.12	2.74
	同一化	<i>-1.94</i>	<i>-1.48</i>	-2.37	0.30
	取り入れ	0.45	<i>-1.17</i>	<i>-1.62</i>	0.16
	外的	0.28	0.62	<i>-1.51</i>	-0.62
緑	内的	0.76	<i>1.26</i>	2.39	2.79
	同一化	0.78	2.46	<i>-1.10</i>	<i>-1.46</i>
	取り入れ	0.29	0.55	-0.49	-0.54
	外的	<i>1.19</i>	0.71	-2.41	0.70

ネガティブな感情を喚起する場合には赤が効果的であることがわかった。また、学習者ロボットが困惑や理解を示している場合、緑は内的調整スタイルに有効だが、同一化調整スタイルには逆効果になる。

個性分析により、適切なインタラクションは学習者の自己調整スタイルに依存することがわかり、学習者ごとに適応的なインタラクションが可能であることを検討した。こうして、仮説 (B) が証明された。

7.6 検証 (C) : 実際の講義と同様の状況で、Academic Emotions を操作できるか？

7.4 節と 7.5 節では、実際の講義に似せた単一の発話を用いたが、発話の流れや文脈はなかった。しかし、前後の発話の文脈も Academic Emotions に影響を与える可能性がある。そこで、実際の講義を想定し、連続した発話が学習者の Academic Emotions に与える影響について調査した。

7.6.1 概要

Pepper の目の色が学習者の学習活動の感情に与える影響について検討した。実験参加者は大学院生と学部生の計 18 名（実験群 9 名，統制群 9 名）である。

7.6.2 目的

実際の講義において、色彩によって Academic Emotion を操作できるかどうかを調査した。

7.6.3 刺激

7.4 節と同様に、Pepper の目の色だけを変え、目の色と対応する感情は同じにした。統計処理に関する実際の講義を 12 の発話に分け（表 7.6，表 7.7，表 7.8），その発話に重要，

注意、困惑、理解の意図が含まれている場合に色を割り当てた。

7.6.4 質問紙

実験 I と同様に、Academic Emotions を評価項目とした 7 段階のリッカート尺度による質問票を用意した。

7.6.5 手続き

実験では、実際の講義を録画したものをもとに連続的な文脈を作り、Pepper ロボット間のインタラクションを映像で作成した。

実験のセットアップの一例を図 7.7 に示す。参加者はノートパソコンを渡され、ヘッドホン装着した。ノートパソコン上にて Pepper の動画が映し出された (図 7.8)。

動画では、Pepper に先生、TA (Teaching Assistant)、生徒の役割を与えた。教師と TA の役割には、重要や注意を促す発話を、生徒の役割には、困惑や理解を表す発話を使用した。したがって、映像の中では、この 3 つの役割の Pepper がインタラクションを行った。

参加者は、事前に各発言の動画内のやりとりの文脈に慣れ、3 体の Pepper ロボットと同じ環境にいることを想定するよう指示された。そして、各発言を聞いたときに参加者が感じた感情を質問紙で記録した。

18 名の参加者を、実験群 (9 名) と、統制群 (9 名) に分けた。実験群では、Pepper の目の色をシナリオに合わせて変化させた動画を見せ、色と対応する感情表も提示した。統制群では、Pepper の目の色を白に固定した動画を見せ、色と対応する感情表は提示しなかった。

手順は以下の通りである図 7.9。

- 1) 参加者はノート PC の前に座る。
- 2) 参加者は、Pepper の講義行動を動画で評価する。
 - a) Pepper は発話 [1] を発話し、意図があるときは目を光らせる。

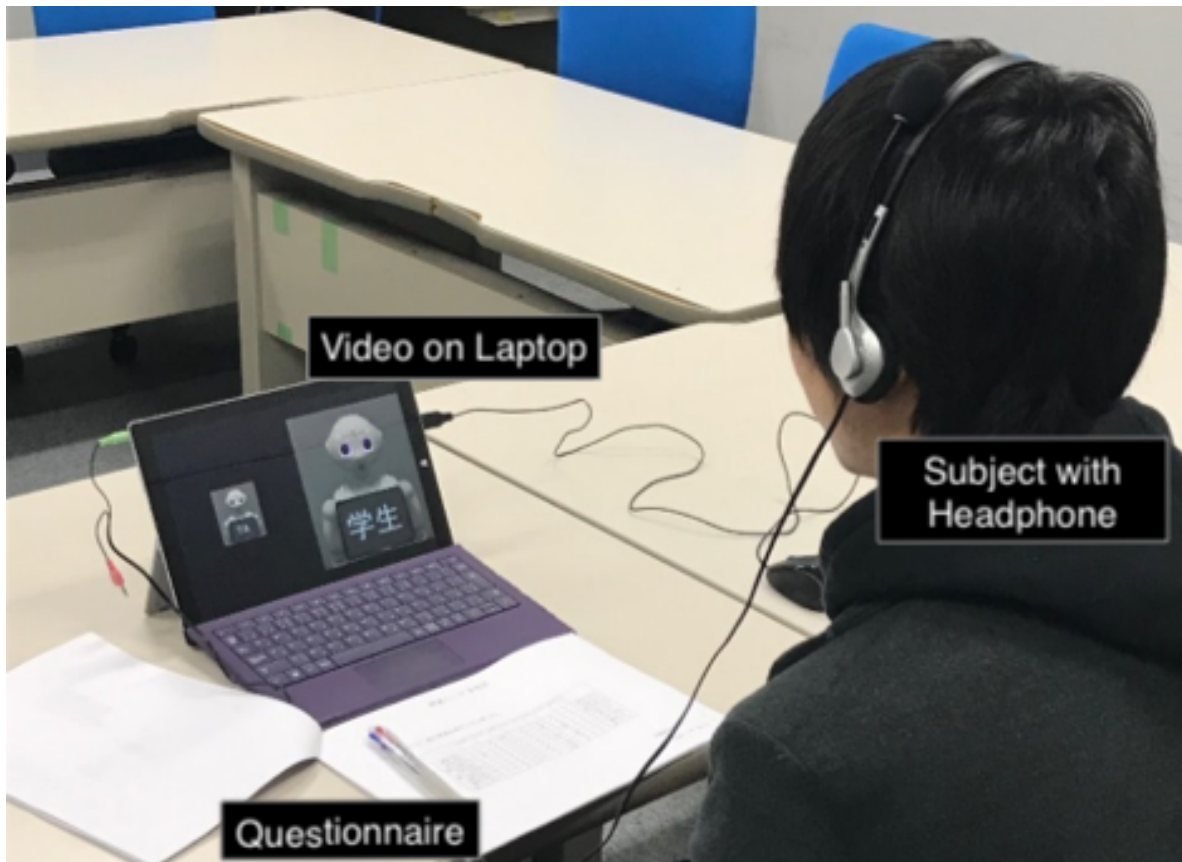


図 7.7 実験設定

- b) 参加者は、アンケートにより文脈で評価する。
- c) 発言に対応する [] の数字を 1~12 まで変えながら、a) と b) を繰り返す。

統制群では、ステップ 2 a) でロボットの目は常に白色で照らされていた。

7.6.6 結果

図 7.10 は実験群の特徴的な結果である発話 2 から 6 の平均値，図 7.11 は統制群の対応する平均値である。

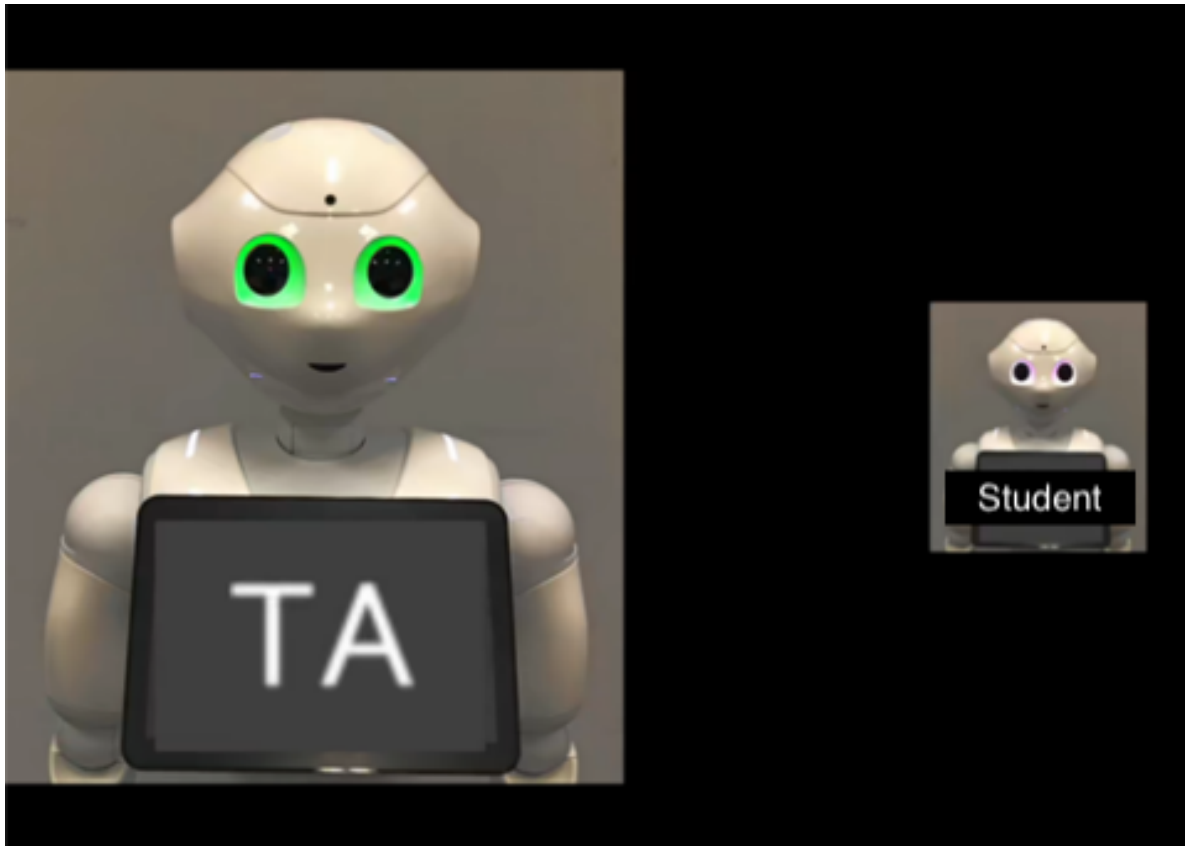


図 7.8 刺激として用いた映像（表 7.3 における発話 10 の状況）

7.6.7 評価

統制群では常に白色であったため、図 7.11 は発話内容そのものに関連する感情を示している。統制群では、連続した発話内容に対する感情の変化はほとんど見られなかったが、実験群（図 7.10）では、複数の感情の変化が見られた。例えば、発話 3 では、実験群の退屈は明らかに増加し、怒りと希望は抑制されている。さらに、発話 5 では、実験群の「誇り」、「怒り」、「希望」が明らかに増加している。これらの結果から、実際の講義に近い状況でも、色によって感情が操作されることが確認された。

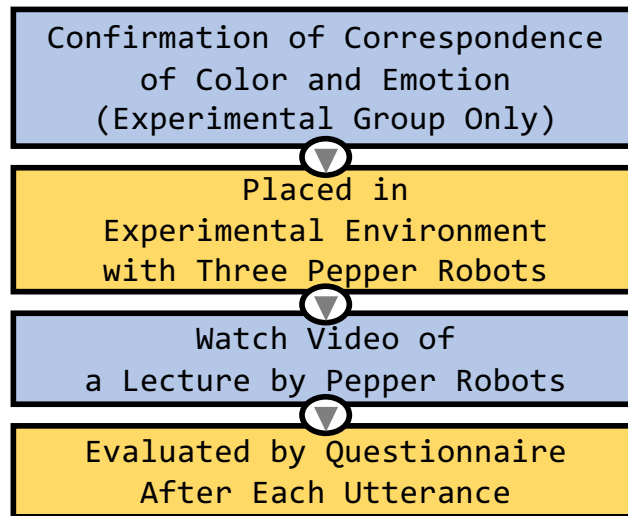


図 7.9 検証 (C) の実験設定

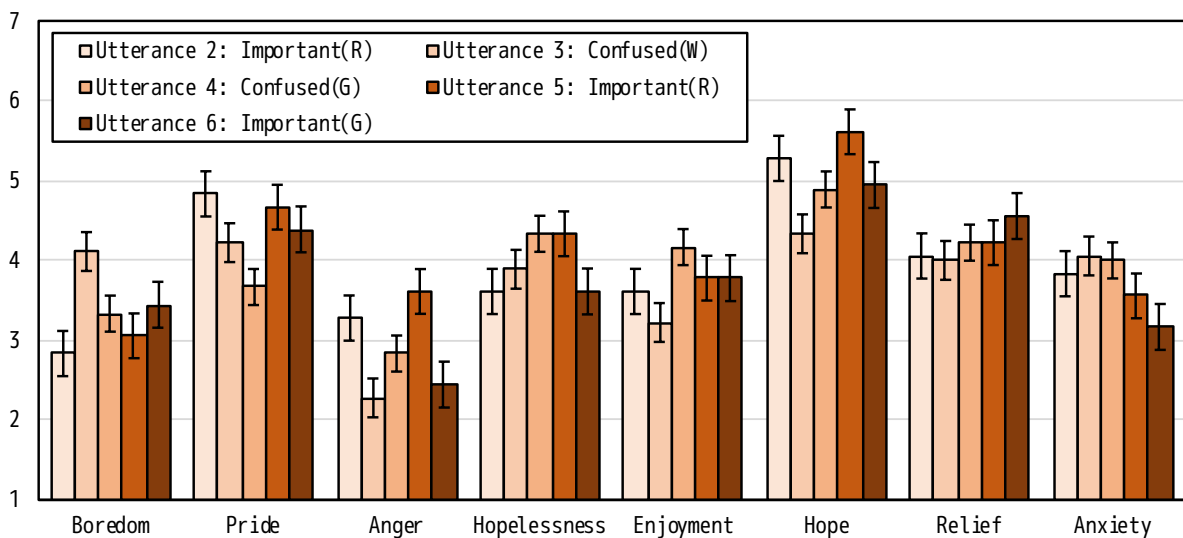


図 7.10 実験群の結果 (エラーバーは標準誤差を示す)

7.7 議論

これらの結果から、ロボットの非言語的行動を用いた教示は学習の場面で用いられてきたが、色を用いた教示は選択肢を増やし、より多様で個人に合わせた教示を提供できることが示唆された。ロボットは人間の精神状態に影響を与えるが、その理由についてはこれまで明確な特徴が示されていなかった。さらに、色彩心理学がロボット主導の教育に応用

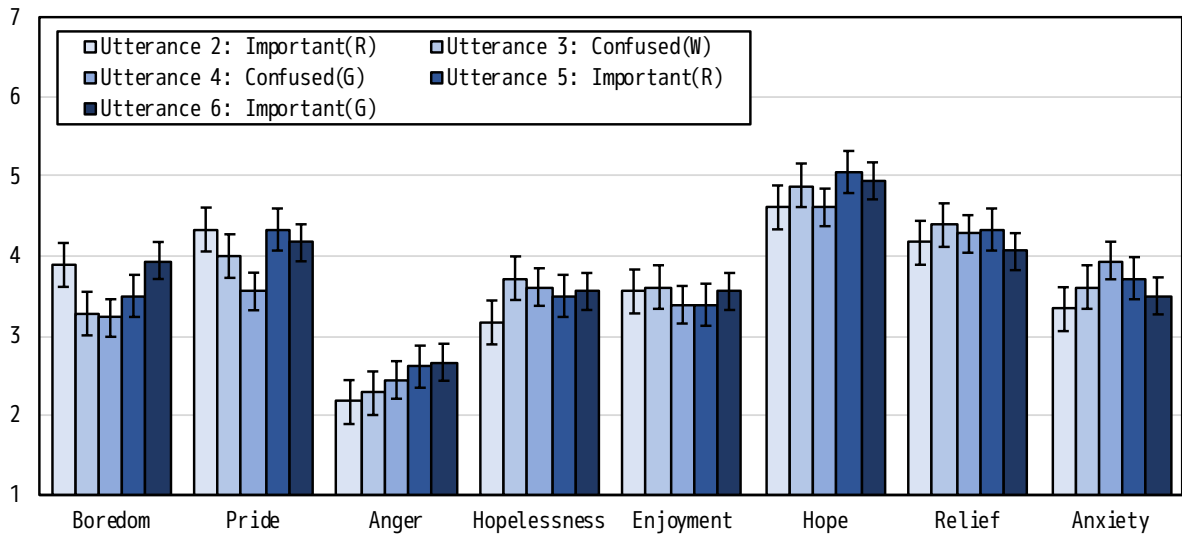


図 7.11 統制群の結果（エラーバーは標準誤差を示す）

できることを示唆している。色彩心理学で長年培われてきた色と感情の関係をロボット主導型教育に用いることは、新しい手法であり、より感情に敏感なロボット主導型教育の開発に役立つと思われる。

7.8 本章のまとめ

ロボットの目の発光色によって喚起される Academic Emotions を調査するために、以下の仮説を検証した。(A) 学習者の Academic Emotions を操作できる、(B) 学習者に適応的な Academic Emotions を生じさせることができる、(C) 実際の講義に近い状況で色により Academic Emotions を操作できる。仮説 (A), (B), (C) の検証はそれぞれ 7.4 節, 7.5 節, 7.6 節で述べたとおりである。

その結果、目の色が Academic Emotion を操作することができ、学習者の個性に応じた適応的な講義行動が生み出されることが示された。さらに、実際の講義に近い状況で、ロボットの目の色によって学究的な感情が操作された。

ただし、これらの結果は限られた発話パターンと色の組み合わせに対するものであり、より詳細な調査が必要である。また、これらの結果は、学習効果ではなく、学習者の感情

に作用することを示している。したがって、今後は、実用現場での再現性の確認と、複数パターンの組み合わせによる汎用性の確認が課題となる。

表 7.6 発話スクリプト

No.	意図	役割	発言 (“[]”内は“()”に示す色を提示)
1	注意	講師	<p>はい、みなさんおはようございます。それでは、本日の講義を始めたいと思います。まず、最初に Q&A の質問と要望のコーナーです。</p> <p>ある学生が、まるまるについて調べよとか、述べよとか、そういうことを、Excel のテキストボックス形式で、やったのですが、これで良いのですか、という質問がありました。</p> <p>これはですね、基本的に課題は、Excel で出すように、ということなので、Excel 中に書いた内容を、保存してください。テキストに書いてもいいですし、セルに直接書いてくても大丈夫です。しかし、今回の提出者の中に、自由記述欄に書いている人が、いました。[(緑) 今回は、仕方がないですが、次からは皆さん、Excel 内に書き込むように、注意してください。]</p>
2	重要	講師	<p>また、統計処理は、前期の授業と比べて、専門性が高くなっているの、難しくなっていると思います。私の講義では、心掛けていることがありまして、単に教えたものを、覚えなさい、というわけではなく、どういったときに使えるのか、ということを中心に、教えています。なので、みなさん、難しいことに慣れましょう。[(赤) また、もう一つ、重要なポイントがありまして、勉強全般に言えることですが、どこが難しいのか、どこが理解しにくいのかという、「どこが」の部分、これをしっかりと考える癖をつけ、どこから分からなくなったのか、ということ、しっかりと、考えてください。]</p>
3	困惑	生徒	<p>[(白) ヒストグラムの作り方で、データ区切り、について、10 から 100 までの間隔というのが、どこにあるのか、もしくは自分で作るのか、わかりませんでした]</p>
4	困惑	生徒	<p>[(緑) 分散と、標準偏差は、求めることができました、しかし、分散や、標準偏差の違いを述べよ、という問題について調べたのですが、違いがよくわかりません]</p>
5	重要	講師	<p>ありがとうございます、これも非常に多かった質問ですね。分散、標準偏差の出し方、違いについても、あとで説明したいと思います</p> <p>皆さん、ヒストグラムや、データ区切りというワードをあげて、どこがわからないのか、ということ、考えられています。</p> <p>実は、使用している講義資料は、前任者のものなので、データ区切りの説明が、なかったと思います。通常、皆さんが手にする資料は、完璧なものはないんです。なので、こういう資料において、補足や、わからない点を、自分なりに調べるといった、スキルを身に付けてほしいと思います。[(赤) なので資料にないことを、説明するさいは、みなさんの方でノートをとる、あるいはメモを取るなど、なんでもいいので、講義資料を補足する活動を行うことが、非常に重要だと思います。]</p>
6	重要	講師	<p>はい次に、「構成累積比の話とか、パレート図の話とかが、結構難しかったです、」というコメントがありました、パレート図や、累積比率は、わかりやすいページを見つければ、すぐに理解できる、簡単なものなんです、わかりづらく書いてあるページのほうが、多いんです。だから、そういうページにあたると、結構、難しいかなと思います。</p> <p>また、「ヒストグラムに対して、累積比率があったほうがいいですか、」という質問がありました、これは、状況によります、だいたい場合は、そこまで要求されないことがあり、ヒストグラムが作ればいい、ということが多いのですが、場合によっては、比率もいっしょに出しておいて、と言われることがあります。[(緑) ここで重要なものが、必要だからやっておいて、と言われたときに、出来るかどうかになります、常にやるかどうかよりも、できるかどうか、重要だということです]</p>

表 7.7 発話スクリプト (続き)

No.	意図	役割	発言 (“[]”内は“()”に示す色を提示)
7	重要	講師	<p>はい、では先ほどお話しした、「分散と、標準偏差の違いが分かりづらいです、」というようなコメントについて説明します。[(赤) この説明は、重要なので、しっかりと聞いてください。] 簡単にいうと、どちらもデータのバラつきを出すためのものです。分散も、標準偏差も、値が小さければ小さいほど、バラつきが少ない、ということになります。</p> <p>それで、違いについてですが、分散の方が、計算が簡単になります。また、分散は、どちらのバラつきが大きいという、比較はできますが、どのくらいバラつきが大きいのか、ということとは分かりません、資料1を見てください。<資料1 についての話>これは、基本統計によって、抽出したのですが、社会と数学、どっちのほうにばらついているか、これは分散によってわかるんですね、しかし、どのくらいばらつきが大きいのか、というのが、分散だとわからないんです、そこで、標準偏差を使います、標準偏差っていうのは、数式がやや難しくなっていますが、平均値から見て、だいたいぜんごに、どのくらいずれているのか、例えば、50点くらいの平均で、ぜんごのばらつきが、5点くらいなのか、50点くらいの平均で、ぜんごのばらつきが、20点くらいなのか、こういうのを求められるのが、標準偏差なんですね、だから、基本的には、標準偏差のほうが、使い道が大きいということです。なので、だいたいの場合は、便利な標準偏差を使うことが多い、というふうに、覚えておくことが、重要になります。</p>
8	重要	講師	<p>はいここまでが、前回の復習になります。[(緑) 話は変わりますが、この講義に限らず非常に大切な話をしたいとおもいます。] 資料2を見てください。<資料2 についての話>この、みなさんの前に表示されている三角形は、ラーニングピラミッドと言われています、これは、大切な話なので、覚えていてほしいんですが、みなさん、大学で一週間くらい講義を聞きますよね、とある研究者が調査したんですが、講義をただ聞く状態だと、知識の定着率が5%、というふうに言われてるんです、だから、私が頑張っても100のことを話しても、5%くらいしか残らない、っていうことになります、なので、このさき、意識してほしいのが、単に講義を聞くだけ、ということはやめてください、これでは、何も身にならないと思います。次に読書をする、本を読むということですね、これだと10%で、ビデオを見る、これだと20%くらい定着すると、さらに実演といった、何かを操作している様子を見ると、30%くらい定着すると言われてます。これらはいずれも、受動的な学習ですね。今、みなさんはどうでしたか？おそらく、受動的な学習をしていたひとが、多いと思います。つまり、自分の将来と、結び付けて考えるといった、意識をはらわない、講義の内容にしか興味がない、といったひとたちは、理解度が低くなりやすいんです。そうではなく、例えば、自分が将来経営者になったとき、こういう自分と関連付けて考えたとき、あるいは人にもものを教える、こういうふうな活動をしたとき、75%から90%、知識が定着している、というふうに言われてるんですね。だから、基本的に知識が定着するのって、インプットしたとき、聞いているときではなく、アウトプットするときなんですね、使うときにしか、知識は定着しない、ということがいわれておりますので、自ら考えていくことが非常に大切なんです、その中でも人に教えていく、ということが、非常に効果が高いので、ぜひ、いろんな講義で、人に教える、ということをやっていただきたいと思います。</p>

表 7.8 発話スクリプト (続き)

No.	意図	役割	発言 (“[]” 内は “()” に示す色を提示)
9	困惑	生徒	[(白) すいません, 入力範囲を指定して, グラフを作るときに, ラベルのチェックを忘れないように, と言っていたのですが, ラベル, というものがどういうものなのか, わかりません]
10	注意	TA	ラベル, というものは, 何かというと, 入力範囲としてどこを選んだか, という時に, 一番上の教科の名前, のところから選んだと思うんだけど, 教科の名前と, いうのは数値じゃないよね, だから, 一番上の教科の名前, というのは数値じゃありませんよ, と宣言するのが, ラベルのチェック, というものになります。ラベルをつけると, 一番先頭はデータではなく, 単なる文字として, 分析してくれます。[(緑) これは, テストの時に間違いやすいので, 注意してください。]
11	理解	生徒	[(赤) なるほど, そういうことなんですね。]
12	理解	講師 & 生徒	<ul style="list-style-type: none"> ・先生ペッパー: 次に平均, 分散, 標準偏差の求め方ですが, この3つは関数で求めることができます。平均は, average, という関数を使います。また, 分散は, VAR, という関数を使い, 標準偏差は, STDEV, という関数を使います。分散と, 標準偏差は, P とか S など, いろいろな種類がありますが, この講義では, どれを用いても構いません。ペッパー学生くんわかりましたか? ・学生ペッパー: [(白) はい, わかりました。]

第 IV 部

領域知識モデルの高次化を伴う 汎用的な拡張法の実現

第 8 章 Error-based Simulation と概念マップを用いたメタ認知駆動学習を促すシステム

8.1 本論文における本章の位置付け

ここまで本論文では、情報構造指向アプローチに基づく ITS の汎化を目指し、第 II 部では情報構造指向アプローチに基づく ITS の対象領域の範囲を未知の対象領域を含む形で広げ、第 III 部では当該の ITS に含まれる領域知識モデルを他の教授モデルにも再利用するための汎用的な手法を示した。よって、より多くの対象領域で情報構造指向アプローチに基づく ITS を開発でき、その開発コストも領域知識モデルの再利用によって低減できることを示してきた。

しかし、情報構造指向アプローチに基づく ITS を開発する際に、必ずしも望ましい領域知識モデルが再利用できる状況とは限らない。このような場合には、新しく 1 から領域知識モデルを構築する方法と、既存の領域知識モデルをベースに一部改変を加えることで望ましい領域知識モデルに変化させる方法が考えられる。後者の方法を更に分解すると、既存の領域知識モデルに対して (1) 複雑化に伴う拡張を加えるパターンと、(2) 高次化に伴う拡張を加えるパターンが考えられる。2 章で議論したように (1) については「加減算から乗除算へ変える」といった拡張が平嶋らの「モンサクン」の範囲ですでに実証されており [山元 17, 平嶋 19]、他の対象領域においても同様の拡張が可能であることが示唆されているといえる。一方で、(2) に挙げた、例えば「加減算から“加減算における自身の問題解決”へ変える」といった学習対象の高次化に伴う拡張は検証されていない。

そこで本章では、近年重要視される学習における学習者自身のメタ認知 (e.g., 自分がどのように考えて問題解決したかを認知して、考えを変更する) への働きかけ [茅島 08, Kayashima 10] に関する領域知識を既存の領域知識モデルに拡張する手法を提案する。これにより、情報構造指向アプローチに基づく領域知識モデルが高次化に伴う拡張が可能であることを実証し、その方法に関する知見を示す。

8.2 メタ認知駆動学習支援としての本章の動機

学習者が自身の経験に基づいて、自身の問題解決過程を振り返り、永続的に調整できるようになるためには「メタ認知駆動学習」が必要とされる [Kayashima 10].

また、Error-based Simulation (EBS) は、学習者の誤りを可視化することによって「学習者自身の認知状態の観察」を促すことが知られている [平嶋 06, Kayashima 10].

しかし EBS 単体では、メタ認知駆動学習において支援不足の箇所が指摘されている [Kayashima 10]. 本章では、EBS に概念マップを用いた解答履歴の可視化・比較活動を加えることでこの解消を指向し、初等力学を対象にメタ認知駆動学習を促すシステムの開発と評価を行う。

8.3 メタ認知とメタ認知駆動学習

8.3.1 関連研究

メタ認知は、問題解決者としての自分に関する知識と、自分の思考過程を客観的に観察し目標に到達するように調整するスキルからなる [茅島 08]. さらに「自分の考え方を客観的に観察し、過去の問題解決を振り返り、学習者が自分の考え方や誤った原因を抽象化できること」をメタ認知活動と呼ぶ [Kayashima 10]. 茅島ら [茅島 08, Kayashima 10] は、メタ認知活動を駆動させることで自身の問題解決過程を自身の手で調整する「メタ認知駆動学習」を提唱している。このメタ認知駆動学習には、「抽象化操作」「修正操作」「具体化操作」の3操作が必要とされる。

抽象化操作とは、自身の問題解決の特徴を言語レベルで抽象化することである。修正操作とは、抽象化した問題解決の特徴を修正、追加、削除することである。また具体化操作とは、抽象化や修正を経た問題解決をインスタンスレベルに具体化することである。

EBS は、「もし学習者の誤った考えが成立するとしたらどのような現象が生起するか」をシミュレートすることで修正を促す学習環境である [Hirashima 98]. EBS では、修正操作と具体化操作を促す一方、抽象化操作の支援不足が指摘されている [Kayashima 10]. 本章ではこの解決を指向することで、メタ認知駆動学習を促す支援を目指す。

8.3.2 本章の位置付け

抽象化操作を促すためには、学習者の問題解決を言語レベルで表現し、それを操作可能な状態で提供することが一つの手段として挙げられる。そこで本章では、従来の EBS の活動における学習者の解答履歴を、言語レベルのラベルとしてマップ形式で概念構造を表現する「概念マップ」に変換し、学習者に提供する。

この言語レベルへの変換には、学習者の解答を概念的に意味付ける過程を要する。例えば、本章で対象とする初等力学では、作図問題として静止物体の図を提示し、その静止物体にかかる力を矢印で記入させる。よって、学習者の解答は図中に記入された矢印である。これを概念マップに変換する際には、各解答に記入された矢印を解釈し、「重力」や「垂直抗力」といったその背景にある知識を補完する必要がある。

このように、解答に対応する概念マップを生成し、学習者に自身が記入した矢印と対応する知識との関係性を意識させる。さらに、概念マップで表現した一連の解答履歴の振り返り活動を加えることにより、各解答間における概念マップの比較を促し、抽象レベルでの問題解決の振り返りや知識の整理、すなわち抽象化操作を支援する。

8.4 提案システム

本章では、初等力学を対象領域として、メタ認知駆動学習の実現を指向した学習支援システムを開発する。8.3.2 で述べたように、提案システムは、EBS を用いた学習支援機能と概念マップを用いた学習支援機能を備え、その 2 つの機能の組み合わせでメタ認知駆動学習を支援する。

まず EBS 機能では、提示された物体に対して、現在働いている力を画面右のフォームを用いて学習者に矢印の形式で記入させる（図 8.1）。その後学習者の解答に対応したシミュレートが行われる。学習者は、学習者の解答に基づくシミュレーションと、正解に基づくシミュレーションの間を比較しながら、正しい解答になるまで試行錯誤を繰り返す。

次に、EBS 機能で正解すると、概念マップ機能に移行する。概念マップ機能では、学習者に対して EBS 機能における解答履歴の一覧を提示する。学習者には、提示された解答履歴から自身の考えが変容したと思われる時点を、解答間のボタンを用いて選択させる。

このボタンを選択すると、その前後の解答に対応した概念マップが表示される（図8.2）。学習者には、その2つの概念マップを比較させる。

表示される概念マップでは、解答の矢印に基づいて、クリックを用いて選択可能な概念ノードの接続関係が提示される（図8.2 右側拡大部）。また、矢印の有無によって表示される概念ノードの数が変わり、矢印の起点や終点、大きさ、対象の物体によってもマップが変化することになる。

概念マップの比較において表示される概念マップは、ある時点 t の解答に対応する概念マップ M_t と、次の時点 $t+1$ の解答に対応する概念マップ M_{t+1} である。よって、学習者に両者の概念マップを比較させる際には、時点 t から $t+1$ にかけて概念ノードが増えていれば、 M_{t+1} 側の概念マップ上で、 M_t には無い概念ノードを選択させる。また、時点 t から $t+1$ にかけて概念ノードが減っていても、 M_t 側の概念マップ上で、 M_{t+1} には無い概念ノードを選択させる。

また、学習者が解答間のボタンを選択した際、その解答間以前に概念ノードの差分がある解答間をシステムが診断し、そのような解答間があれば、そちらを先に探索するように

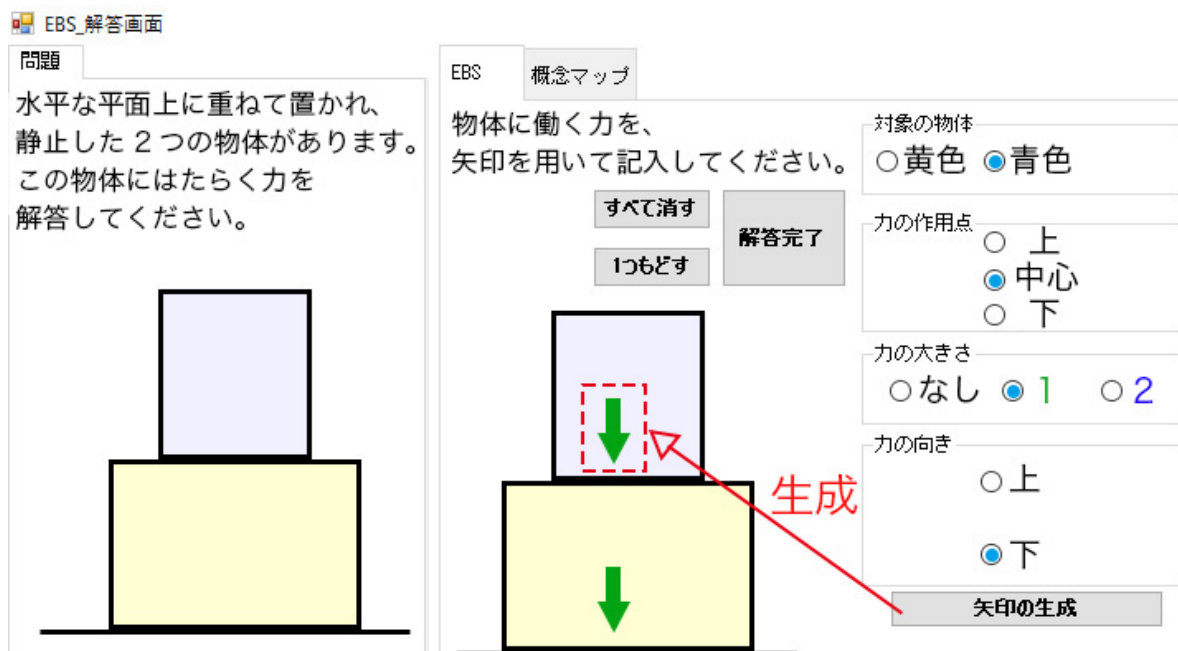


図 8.1 提案システムにおける EBS 機能画面

促す（図 8.2 中央ダイアログ）。この診断を実現するために、システムは EBS 機能から概念マップ機能へ移る際に、学習者の解答履歴にある全ての解答に対して、矢印と概念ノードの対応付けを行い、概念マップの表示に要するデータを生成および記録している。さらに、繰り返し処理を用いて、それぞれの解答間にある概念ノードの差分を記録している。例えば、 M_t と M_{t+1} の差分、 M_{t+1} と M_{t+2} の差分、 \dots 、といったように記録する。この記録を参照し、システムは差分のある解答間を診断する。

概念マップ機能では、このようにして、「自身の思考の変化」についての思考を支援する。

8.5 評価実験

8.5.1 手続き

大学生 17 名を対象に提案システムの評価を行った。

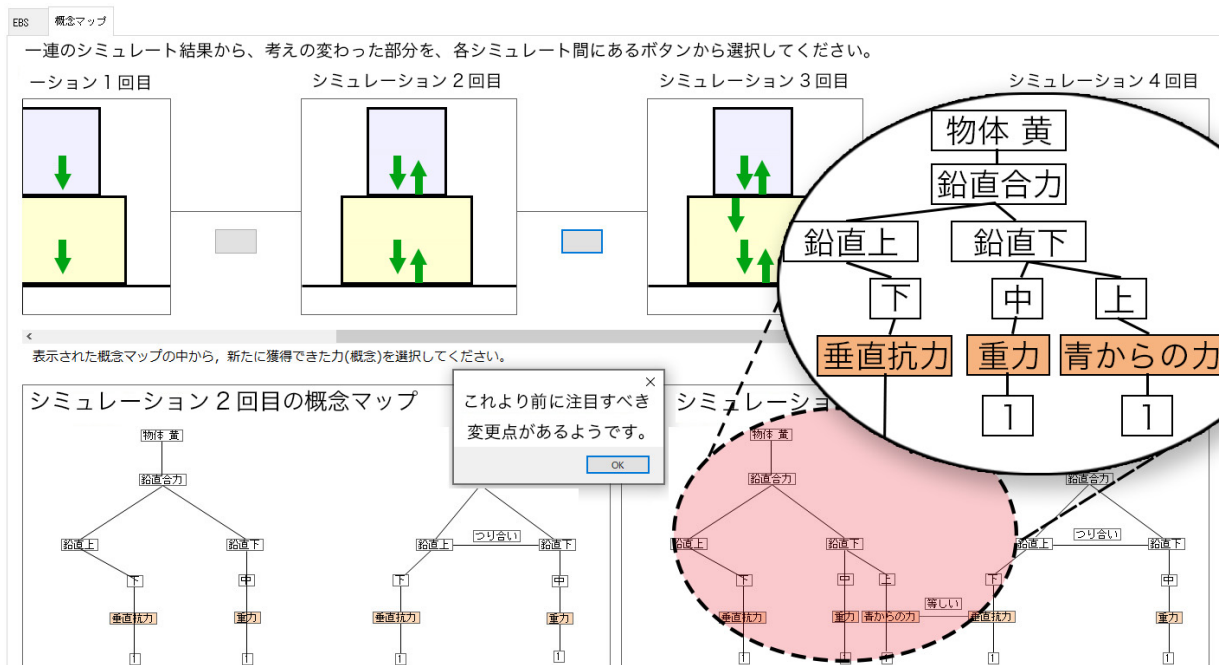


図 8.2 提案システムにおける概念マップ機能画面

まず、提案システムを用いる実験群 9 名と、従来 EBS システムを用いる統制群 8 名に分け、20 分間システムによる学習を行った。その後、振り返り記述紙を 10 分間にて行い、最後にアンケートを実施した。

振り返り記述紙では、各システムが実験参加者の抽象化操作を促すかを測るため、自由記述形式で「自分の考えが変化したポイント」「どの力に対する考え方が変化したのか」「どのような考え方に変化したのか」を、各システムを自由に触りながら記述してもらった。

8.5.2 振り返り記述紙の結果

振り返り記述紙における特徴的な記述を抜粋する。実験群の結果から「2 回目と 3 回目のシミュレーション間で、黄色の物体のみに力を書いても間違っていたから、青にも同じように力を追加した」という記述や、「3 回目と 4 回目のシミュレーション間で、上より下の物体のほうが力が加わっていると考えて、下の物体の力を 2 倍にした」といった記述がみられた。また、どの実験参加者も 1 回目から順にシミュレーションを振り返っていた。よって、提案システムが学習者自身の解答履歴を辿らせ、抽象化操作を促すことが示唆された。

一方、統制群の結果では自分の考えが変化したポイントや、どの力に対する考え方が変化したのかが記述されていなかった。よって、従来 EBS だけでは自分の解答履歴を振り返ることが困難であると分かった。

8.5.3 アンケート結果

アンケートの結果（表 8.1）から、ID1-3 では、実験群と統制群で平均に差がみられることから、提案システムの利用が、獲得したものや修正できた誤りを理解すること、過去の解答の振り返りを促したと考えられる。実験群のみに質問した ID4-6 に関しては、いずれも高い評価が得られ、提案システムが“自分の中の概念への理解状態の認識”や“自分の考えの整理”、“獲得した概念への意識”を促すことが示唆された。

8.6 本章のまとめ

本章では，従来の EBS に加え概念マップ同士の比較活動を組み合わせることで，抽象化操作まで含めたメタ認知駆動学習を促すシステムの開発と評価を行った．評価からは，提案システムが学習者へ抽象化操作を含めたメタ認知活動を促す可能性が示唆された．

表 8.1 アンケートにおける質問項目（6 件法）

ID	質問項目	実験群	統制群
1	システムは自分が獲得したものを理解するのに有効であると思うか	4.89	3.75
2	システムは修正できた誤りを理解するのに有効であると思うか	5.33	4.13
3	システムが過去の解答を振り返るのに有効であると思うか	5.56	2.88
4	システムの概念マップ機能は自分の中にある概念の理解状態を認識するのに有効であると思うか	4.78	—
5	システムの概念マップで自分の考えを整理することができると思うか	5.11	—
6	新たに獲得した概念が何であるかを意識することにつながると思うか	5.00	—

第 V 部

総括

第9章 本論文の貢献と限界

本章では、本論文の貢献と限界についてまとめる。まず、各部の貢献を改めてまとめ、それらを踏まえて、全体における貢献について述べる。最後に、本論文の限界についてまとめる。

第II部では、未知な領域に対する情報構造アプローチの適用を実現し、その方法に関する知見を示した。具体的には、知識の組織化を目的とした教授モデルが適用可能であり、機能-振る舞い-構造で知識が表現できる対象においては、未知な領域であっても類似の方法で情報構造指向アプローチに基づくITSが開発できることを示唆した。

第III部では、既存の情報構造指向アプローチに基づくITSに含まれる領域知識モデルの再利用を実現し、その方法に関する知見を示した。具体的には、既存の領域知識モデルを汎用的に再利用できる教授モデルとして Learning-by-teaching を支援する教授モデルを実現し、その事例から、どのような教授モデルであれば、また、どのような領域知識モデルであれば再利用しうるかを示唆した。

第IV部では、既存の情報構造指向アプローチに基づくITSに含まれる領域知識モデルの高次化に伴う拡張を実現し、その方法に関する知見を示した。具体的には、既存の領域知識モデルに対して、そのモデルで対象となる領域における学習者自身の問題解決そのものに関する領域知識の拡張を実現し、その方法を示した。

これらをまとめると、本論文全体の貢献としては、情報構造指向アプローチに基づくITSが、プログラミング等のより未成熟かつ複雑な領域においても構築でき、構築した領域知識モデルが一定の情報を持っていれば他の教授活動に再利用可能であり、さらに状況に応じて領域知識モデルの複雑化だけでなく、高次化に伴う拡張・変更が可能であることを事例的に示すと同時に、これらを実現するための方法論に関する知見を示唆したといえる。展望としては、本論文で示された各要素技術はいずれも対象領域から独立でき、1つのITSとして融合しうると考えている。

また、本論文では、情報構造指向アプローチに基づくITSの汎化を目指し、そのアーキテクチャをどのように発展させていくかに主眼を置いて議論を進めた。その一方で、ITS開発を行う際には、「そもそもどのような学習活動を実現ないし支援しようとしているか」

が前提として非常に肝要となる。本論文では各部それぞれによって特定のコンテキストでは新たな学習活動の提案を含めて ITS を設計している。よって、いくつかの点で、領域知識モデルを再利用する場合に活用できる学習活動の設計手法に関する知見を示せていると考えている。具体的には、第 II 部で得られた知見からは、「知識に 2 種以上の表現を持つあらゆる学習対象 (e.g., ソースコードと機能, 数式とグラフ, 物理方程式と振舞い)」に対して、その表現のセットを知識として組織化するための学習活動を設計できる汎用的手法が示されている。また、第 III 部で得られた知見から、領域知識モデルの「解答に用いる構成要素」と「正答で満たすべき構成要素間の制約」を用いることで、領域独立かつ領域知識モデルの特性を潰すことなく新たな学習活動が設計できることが示されている。さらに、第 IV 部で得られた知見からは、領域知識モデルが持つ解答に対する構造を学習者に概念マップとして公開および活用することで、より高次の学習活動が設計できることが示されている。

一方、本研究の限界としては、各部で示唆された知見の汎用性が実証されていないことが挙げられる。第 II 部、第 III 部、第 IV 部で取り組んだ各課題については事例的に解決したのみであり、他の対象においても実現しうるかは未検証である。しかしながら、いずれもこれまで実現されていなかったことについて、実現しうるということを事例的に示したのみでも一定の価値があると考えている。また、「ITS においてどのように学習活動を設計すべきか」という核心に迫る議論には至っていないため、今後の追究が必要である。

第 10 章 結論

本論文では情報構造指向アプローチに基づく ITS の汎化を目的として、以下に述べる 3 つの課題を設定し、多角的な発展を試みた。

第 II 部 未知の対象領域に対する領域知識モデル構築法の実現: 未成熟かつ複雑な対象領域における領域知識モデル構築法に関する知見創出

第 III 部 領域知識モデルの汎用的な再利用法の実現: 既存の領域知識モデルを再利用可能かつドメイン独立な教授モデルの実現と、モデルの再利用法および再利用可能なモデルについての知見創出

第 IV 部 領域知識モデルの高次化を伴う汎用的な拡張法の実現: 既存の領域知識モデルにおける学習対象の高次化を伴う拡張法に対する知見創出

1 つ目の課題に対し、第 II 部では、従来研究の対象領域と比較してより未知な対象領域、すなわち未成熟かつ複雑な対象領域における領域知識モデル構築法に関する知見を創出することを目的とした研究について述べた。具体的には、3 章では、プログラミングにおける領域知識モデルを情報構造指向アプローチに基づいて構築するために、機能的階層構造の理解を目的とした学習活動を新たに設計した [古池 19a]。これにより、任意の機能的な階層構造を学習対象化でき、教授者が指定する任意の粒度での領域知識モデル構築を可能とした。さらに 4 章では、3 章で提案した学習活動に基づき、それに付随する問題解決過程について議論した [古池 20]。この議論を通して、提案した学習活動において必要となる知識かつ結果として獲得される知識の構造を情報構造として特定し、その情報構造を用いて問題解決過程を計算機可読なモデルとして提案した。これに付随して、計算機が問題解決過程を把握するのに用いる情報構造も自明となり、プログラミング学習における領域知識モデルの一つを明らかにした。また、5 章では、4 章で問題解決過程において必要となる知識構造を明らかにしたことで、学習者自身の知識習熟度を表現する学習者モデルを提案し、その学習者モデルに基づいて 3 章で提案したシステムの振舞いを適応的に変化しうることを確認した [古池 23b]。これにより、情報構造指向アプローチに基づいた領域知識モデルをプログラミング領域で構築しうることを、一連の研究を通して確認し、

未知領域において領域知識モデルを構築する方法の知見を示した。

2つ目の課題に対し、第III部では、既存の領域知識モデルを再利用可能かつドメイン独立な教授モデルを実現し、モデルの再利用法および再利用可能なモデルについての知見を得ることを目的とした研究について述べた。具体的には、6章では、既存の学習支援システムにおける学習対象および学習活動に関するモデルを再利用することで Teachable Agent を追加実装できるフレームワークを実現した [古池 23a]。同時に、領域知識モデルとして任意の構成要素とその構成要素間の制約が記述されていれば、他の教授モデルに組み合わせを変更できることを示した。さらに7章では、Teachable Agent の社会的存在らしさが学習効果に影響すること、社会的存在らしさを向上させるためにはロボット化や感情表出が一つの手段であることを挙げ、Teachable Agent をロボット化し、さらに学習者の感情制御する手法について検討した [Koike 19b]。これにより、発話文脈とロボットの目の発色の組み合わせによって学習者の感情を制御する手法を提案し、学習者に対する一定の感情の制御可能性を明らかにした。よって、6章と7章の融合については仮説的であるもののほぼ自明であり、情報構造指向アプローチに基づく既存領域知識モデルを、感情的支援を含むかたちで他の教授モデルに再利用する方法に関する知見を一連の研究から示した。

3つ目の課題に対し、第IV部では、既存の領域知識モデルにおける学習対象の高次化を伴う拡張法に対する知見の創出を目的とした研究について述べた。具体的には、8章において既存の学習支援システムを対象とした、メタ認知活動の促進を指向した追加機能の実現に関する研究を行った [古池 21]。これにより、情報構造指向アプローチに基づく既存領域知識モデルに対して高次化に伴う拡張を加える方法に関する知見を事例的に示した。

このように本論文では、情報構造指向アプローチに基づく ITS の汎化を目的として多角的な発展を試みた。これらの成果によって、情報構造指向アプローチに基づく ITS が、プログラミング等のより未成熟かつ複雑な領域においても構築でき、構築した領域知識モデルが一定の情報を持っていれば他の教授活動に再利用可能であり、さらに状況に応じて領域知識モデルの複雑化だけでなく、高次化に伴う拡張・変更が可能であることを事例的に示すと同時に、これらを実現するための方法論に関する知見を示唆したといえる。

最近では、情報構造指向アプローチに基づく ITS はより一層の広がりを見せている。例

例えば岩井らは、日本語（国語）学習において、比較的一意に解を決定しうる「異議可換語」に絞って学習対象化することで、情報構造指向アプローチに基づく領域知識モデルの構築およびその ITS の実現に成功している [岩井 21]。また、相川らは、情報構造指向アプローチに基づく古典力学（ニュートン力学）を対象とした ITS である Error-based Simulation (EBS) システムを対象に、領域知識モデルにおいて課題間の関係性を表現することで、任意の誤りに対応する補助課題推薦を可能としている [相川 20, 相川 23b]。一方、EBS システムの教授モデルを拡張することで、Ueno らによる学習者が提示された物体に働く力を観測するための手法 [Ueno 19] や、相川らによる学習者が提示した誤答を肯定した場合の物理状況を可視化する手法 [相川 23a] が提案されている。さらに植野らは、EBS システムの教授モデルを変更することで、学習者が提示された EBS システムの誤答例に対して存在する誤りを指摘するための手法 [植野 20] を提案している。また、本論文で提案した領域独立な教授モデルである TAME を古典力学に対して適用を試みた事例 [荻田 22] や、同じく本論文で提案したプログラミングの領域知識モデルを用いてプログラムの読解支援の教授モデルへの活用を試みた事例 [松為 22] も既に存在している。これらの研究を含めて、情報構造指向アプローチに基づく ITS の汎化については今後も事例的に発展させていく必要があり、依然追求の余地があるといえる。

また、9章で示した他にも、いくつかの general な議題が残されている。例えば、Ohlsson が非等結果性の原則（The Principle of Non-Equifinality）として指摘 [Ohlsson 86] するように、学ぶ対象となる領域知識のモデリングに模範的な方法はなく、どのような知識も無数の異なる観点から表現することができる。また、このような ITS の各コンポーネントの教育的妥当性の評価の必要性 [Mark 93] は古くから指摘されてきている。したがって、本論文の一連の研究で示されたモデルが、他の観点からモデル化する場合と比べて相対的に教授的妥当性が高いかどうかの議論は難しく、今後も継続的な実践と慎重な分析を通して検証していく必要があるといえる。

謝辞

本論文もしくは関連する研究を遂行するにあたり、非常に多くの方々にご指導、ご助言、ご協力賜りましたことを心より感謝申し上げます。

特に、指導教員である東京工芸大学工学部 東本崇仁准教授には、本論文の主査だけでなく、著者が東京工芸大学に在学した9年間のうち、その大半を占める8年の長きにわたって、格別のご指導を賜りました。筆者が氏に対して抱く感謝の念は筆舌に尽くし難く、氏の寛大かつ本質的な指導のもとで、研究者としてだけではなく、一人の人間として何段階も成長する機会を与えて頂きました。筆者が学部2年生のときに本学に着任した氏との偶然的な巡り合わせがなければ、著者が博士課程に進学することも、博士課程を全うすることも叶いませんでした。また、氏との間で交わされた数多の議論の全てが、今の筆者を形成しており、筆者が研究生活を推進する原動力となり、筆者が研究者を志す理由となりました。衷心より御礼申し上げます。

本論文の副査である東京工芸大学工学部 上倉一人教授、東京工芸大学工学部 片上大輔教授、東京工芸大学工学部 姜有宣教授には、度重なるご指摘・ご教示を頂き、本論文を良いものとする機会を与えていただきました。また、同じく副査を外部から務めて頂いた広島大学大学院先進理工学研究科 平嶋宗教授、東京理科大学工学部 赤倉貴子教授には、学習工学および教育工学の専門的見地から、本論文の方向性に対するご教示を頂き、本論文をより良いものとする機会を与えていただきました。大変感謝申し上げます。

また、本論文の主要な研究および関連する研究の共著者として、広島大学大学院先進理工学研究科 平嶋宗教授、神戸大学大学院海事科学研究科 堀口知也教授の両氏には、的確なご指導およびご教示を頂き、また、活発な議論をさせて頂きました。それだけでなく、著者は学部生の頃から博士課程に至るまで、両氏が博士課程のときから積み上げてきた学習工学に関わる数々の先駆的な功績に対して何度も魅力を感じ、興奮を禁じえませんでした。両氏の数々のアイデアに魅了されたことが、著者がこの分野を志す最大の理由となりました。ここに深くお礼申し上げます。

そして、学習工学研究室の設立年度より歴代の皆様には、日頃より研究についての議論、ご助言、ご協力を賜るだけでなく、日常生活においても数多くの時間を共有していた

いただきました。中でも、半分以上もの時間を共にした、植野和氏、相川野々香氏、茂木誠拓氏には、筆者の研究室生活に大きな刺激を与えて頂きました。また、黒川魁氏、前田新太郎氏にも、様々な場面でお世話になりました。辻優也氏、久世泰成氏、樋村いづみ氏、藤島優希氏には、筆者と共に本論文に関わる研究を推進して頂きました。そのほかにも、ここで挙げきることは叶いませんが、沢山の方々と研究や研究生活を共にし、お世話になりました。心より感謝申し上げます。

筆者の研究生生活の一部は、Intelligent Tutoring System 勉強会、教育・学習支援システム若手の会、日本感性工学会学生会「志学の会」といった若手団体の活動およびそのメンバーによっても支えられました。特に、大阪府立大学（現：大阪公立大学）油谷知岐氏、早稲田大学 田和辻可昌講師、近畿大学 山元翔講師には、研究生生活の刺激となるような意見交換をさせていただき、大変お世話になりました。感謝申し上げます。

株式会社 manabo および従業員の皆様には、「学業を優先しながら働きたい」という筆者のわがままを快く受け入れて頂き、正社員として種々のご配慮を賜りながら1年間働かせていただきました。それだけでなく、本分野の学術と社会の関係性について知りたいという筆者の興味のもと、教育システム産業について多くのことを学ばせて頂きました。大変感謝申し上げます。

友人である、桑野遼氏、深山颯斗氏、米川亮氏、梅田知晃氏には、これまでの私生活の多くを共有していただき、研究生生活を続ける活力になりました。また、家族である、沙樹、加枝、慎太郎、兼幸、麻美、達也、直樹、唯人の存在は、様々な場面で心の支えとなりました。特に、妻の沙樹には、私生活の全ての場面とって過言ではないほどの場面で、筆者の研究生生活に対して多大な支援をして頂きました。また、母の加枝には、ここまで筆者を育てるだけでなく、その寛容な心から筆者の自由な行動を暖かく見守っていただき、また、筆者の研究生生活を色々な側面から応援して頂きました。

最後に、本論文を、筆者が研究の道へ進む大きなきっかけと動機付けを与えてくれた、祖母の故 大森昭恵氏に捧げます。

（初版） 2022年9月3日 千葉県船橋市の自宅にて

（改訂） 2023年1月30日 神奈川県厚木市の研究室の居室にて

古池 謙人

参考文献

- [Adelson 81] Adelson, B.: Problem solving and the development of abstract categories in programming languages, *Memory & Cognition*, Vol. 9, pp. 422–433 (1981)
- [相川 20] 相川野々香, 古池謙人, 東本崇仁: Error-based Simulation (EBS) における学習者の誤り傾向に基づく問題の行詰まり解決支援システム, 電子情報通信学会論文誌 D, Vol. 103, No. 9, pp. 644–647 (2020)
- [相川 23a] 相川野々香, 齊藤寛, 古池謙人, 東本崇仁: 力学における学習者の誤りに基づいた問題提示 (Error-based Problem Posing) システムの開発, 電子情報通信学会論文誌 D, Vol. J106-D, p. in press (2023)
- [相川 23b] 相川野々香, 古池謙人, 東本崇仁, 堀口知也, 平嶋宗: 力学を対象とした Error-based Simulation における行き詰まりの解消を指向した補助問題の提示システム, 電子情報通信学会論文誌 D, Vol. J106-D, p. in press (2023)
- [Belpaeme 18] Belpaeme, T., Kennedy, J., Ramachandran, A., Scassellati, B., and Tanaka, F.: Social robots for education: A review, *Science robotics*, Vol. 3, p. eaat5954 (2018)
- [Biggerstaff 87] Biggerstaff, T. and Richter, C.: Reusability Framework, Assessment, and Directions, *IEEE Software*, Vol. 4, pp. 41–49 (1987)
- [Bimba 17] Bimba, A. T., Idris, N., Al-Hunaiyyan, A., Mahmud, R. B., and Shuib, N. L. B. M.: Adaptive feedback in computer-based learning environments: a review, *Adaptive Behavior*, Vol. 25, pp. 217–234 (2017)
- [Biswas 05] Biswas, G., Leelawong, K., Schwartz, D., Vye, N., and Vanderbilt, at T. T. A. G.: Learning by teaching: A new agent paradigm for educational software, *Applied Artificial Intelligence*, Vol. 19, No. 3-4, pp. 363–392 (2005)
- [Carbonell 70a] Carbonell, J. R.: AI in CAI: An artificial-intelligence approach to computer-assisted instruction, *IEEE transactions on man-machine systems*, Vol. 11, No. 4, pp. 190–202 (1970)
- [Carbonell 70b] Carbonell, J. R.: *Mixed-initiative man-computer instructional dia-*

- logues*, PhD thesis, Massachusetts Institute of Technology (1970)
- [Chase 73] Chase, W. G. and Simon, H. A.: Perception in chess, *Cognitive psychology*, Vol. 4, pp. 55–81 (1973)
- [Chase 09] Chase, C. C., Chin, D. B., Oppezzo, M. A., and Schwartz, D. L.: Teachable agents and the protégé effect: Increasing the effort towards learning, *Journal of Science Education and Technology*, Vol. 18, No. 4, pp. 334–352 (2009)
- [Chrysafiadi 13] Chrysafiadi, K. and Virvou, M.: Student modeling approaches: A literature review for the last decade, *Expert Systems with Applications*, Vol. 40, pp. 4715–4729 (2013)
- [Corbett 94] Corbett, A. T. and Anderson, J. R.: Knowledge tracing: Modeling the acquisition of procedural knowledge, *User Modeling and User-Adapted Interaction*, Vol. 4, pp. 253–278 (1994)
- [De Kleer 84] De Kleer, J. and Brown, J. S.: A qualitative physics based on confluences, *Artificial Intelligence*, Vol. 24, pp. 7–83 (1984)
- [Dennis 16] Dennis, M., Masthoff, J., and Mellish, C.: Adapting Progress Feedback and Emotional Support to Learner Personality, *International Journal of Artificial Intelligence in Education*, Vol. 26, pp. 877–931 (2016)
- [Deublein 18] Deublein, A., Pfeifer, A., Merbach, K., Bruckner, K., Mengelkamp, C., and Lugin, B.: Scaffolding of motivation in learning using a social robot, *Computers and Education*, Vol. 125, pp. 182–190 (2018)
- [D’Mello 07] D’Mello, S., Picard, R. W., and Graesser, A.: Toward an affect-sensitive AutoTutor, *IEEE Intelligent Systems*, Vol. 22, pp. 53–61 (2007)
- [Duran 17] Duran, D.: Learning-by-teaching. Evidence and implications as a pedagogical mechanism, *Innovations in Education and Teaching International*, Vol. 54, No. 5, pp. 476–484 (2017)
- [江木 09] 江木鶴子, 竹内章: プログラミング初心者にはトレースを指導するデバッグ支援システムの開発と評価, *日本教育工学会論文誌*, Vol. 32, pp. 369–381 (2009)
- [Elliot 07] Elliot, A. J. and Maier, M. A.: Color and psychological functioning, *Cur-*

- rent Directions in Psychological Science*, Vol. 16, pp. 250–254 (2007)
- [Exton 02] Exton, C.: Constructivism and program comprehension strategies, in *Proceedings - IEEE Workshop on Program Comprehension*, Vol. 2002-Janua, pp. 281–284, IEEE (2002)
- [Fiorella 13] Fiorella, L. and Mayer, R. E.: The relative benefits of learning by teaching and teaching expectancy, *Contemporary Educational Psychology*, Vol. 38, No. 4, pp. 281–288 (2013)
- [Gao 06] Gao, X. P. and Xin, J. H.: Investigation of human’s emotional responses on colors, *Color Research and Application*, Vol. 31, pp. 411–417 (2006)
- [Graesser 99] Graesser, A. C., Wiemer-Hastings, K., Wiemer-Hastings, P., Kreuz, R., and Group, T. R.: AutoTutor: A simulation of a human tutor, *Cognitive Systems Research*, Vol. 1, pp. 35–51 (1999)
- [Gulz 20] Gulz, A., Londos, L., and Haake, M.: Preschoolers’ understanding of a teachable agent-based game in early mathematics as reflected in their gaze behaviors—an experimental study, *International Journal of Artificial Intelligence in Education*, Vol. 30, No. 1, pp. 38–73 (2020)
- [Harth 17] Harth, E. and Dugerdil, P.: Program understanding models: An historical overview and a classification, in *ICSOFT 2017 - Proceedings of the 12th International Conference on Software Technologies*, pp. 402–413, SciTePress (2017)
- [Hasanah 17] Hasanah, N., Hayashi, Y., and Hirashima, T.: An analysis of learner outputs in problem posing as sentence-integration in arithmetic word problems, *Research and Practice in Technology Enhanced Learning*, Vol. 12, No. 9, pp. 1–16 (2017)
- [Hasanah 19] Hasanah, N., Hayashi, Y., and Hirashima, T.: Posing Arithmetic Word Problems in a Sentence Integration Learning Environment in English and Indonesian: A Utilization Analysis, *The Journal of Information and Systems in Education*, Vol. 18, No. 1, pp. 51–62 (2019)
- [林 17] 林 雄介, 平嶋宗: 学習内容の情報構造化と構造操作タスクの設計に基づく

- Moderately-defined な問題による教科学習におけるプログラミング的思考の部分
タスク化, 人工知能学会第 81 回先進的学習科学と工学研究会, pp. 12–19 (2017)
- [Heffernan 08] Heffernan, N. T., Koedinger, K. R., and Razzaq, L.: Expanding the
model-tracing architecture: A 3rd generation intelligent tutor for algebra symbol-
ization, *International Journal of Artificial Intelligence in Education*, Vol. 18, pp.
153–178 (2008)
- [Hicks 14] Hicks, A., Peddycord, B., and Barnes, T.: Building games to learn from
their players: Generating hints in a serious game, in *International Conference on
Intelligent Tutoring Systems*, pp. 312–317, Springer (2014)
- [平嶋 92] 平嶋宗, 中村祐一, 池田満, 溝口理一郎, 豊田順一: ITS を指向した問題解決モ
デル MIPS, 人工知能学会誌, Vol. 7, pp. 475–486 (1992)
- [平嶋 95] 平嶋宗, 東正造, 柏原昭博, 豊田順一: 補助問題の定式化, 人工知能学会誌,
Vol. 10, No. 3, pp. 413–420 (1995)
- [Hirashima 98] Hirashima, T., Horiguchi, T., Kashihara, A., and Toyoda, J.: Error-
Based Simulation for Error-Visualization and Its Management, *International Jour-
nal of Artificial Intelligence in Education*, Vol. 9, No. 1-2, pp. 17–31 (1998)
- [平嶋 99] 平嶋宗: 学習者モデリングと適応的インタラクション, 人工知能学会誌, Vol. 14,
pp. 17–24 (1999)
- [平嶋 06] 平嶋宗: メタ認知の活性化支援, 人工知能, Vol. 21, pp. 58–64 (2006)
- [平嶋 13] 平嶋宗: 学習課題の内容分析とそれに基づく学習支援システムの設計・開発: 算
数を事例として, 教育システム情報学会誌, Vol. 30, No. 1, pp. 8–19 (2013)
- [Hirashima 14] Hirashima, T., Yamamoto, S., and Hayashi, Y.: Triplet structure
model of arithmetical word problems for learning by problem-posing, in *HCI Inter-
national 2014*, pp. 42–50, Springer (2014)
- [平嶋 15] 平嶋宗: 「学習課題」 中心の学習研究: 情報構造としての学習課題の再定義と構
造操作としての学習活動の設計, 人工知能, Vol. 30, No. 3, pp. 277–280 (2015)
- [Hirashima 16] Hirashima, T. and Horiguchi, T.: Creation of Cognitive Conflict by
Error-Visualization: Error-Based Simulation and Its Practical Use in Science Learn-

- ing, in Spector, M. J., Lockee, B. B., and Childress, M. D. eds., *Learning, Design, and Technology: An International Compendium of Theory, Research, Practice, and Policy*, pp. 1–33, Springer International Publishing (2016)
- [Hirashima 17] Hirashima, T., Shinohara, T., Yamada, A., Hayashi, Y., and Horiguchi, T.: Effects of Error-Based Simulation as a Counterexample for Correcting MIF Misconception, in *Proceedings of the 18th International Conference on Artificial Intelligence in Education*, pp. 90–101 (2017)
- [平嶋 18] 平嶋宗, 林雄介: メタ問題設計法としてのオープン情報構造アプローチ, 人工知能学会第 82 回先進的学習科学と工学研究会, pp. 55–60 (2018)
- [平嶋 19] 平嶋宗: 作問学習に対する知的支援の試みと実践—組立としての作問および診断・フィードバック機能の実現—, 科学教育研究, Vol. 43, No. 2, pp. 61–73 (2019)
- [堀口 08] 堀口知也, 今井功, 東本崇仁, 平嶋宗: Error-based Simulation を用いた中学理科の授業実践—ニュートンの第三法則を事例として—, 日本教育工学会論文誌, Vol. 32, No. Suppl., pp. 113–116 (2008)
- [Horiguchi 14] Horiguchi, T., Imai, I., Toumoto, T., and Hirashima, T.: Error-based Simulation for error-awareness in learning mechanics: An evaluation, *Educational Technology and Society*, Vol. 17, No. 3, pp. 1–13 (2014)
- [Horiguchi 16] Horiguchi, T., Tomoto, T., and Hirashima, T.: The Effect of Problem Sequence on Learners' Conceptual Understanding in Mechanics, *SIG on Advanced Learning Science and Technology*, Vol. 78, pp. 1–5 (2016)
- [堀口 20] 堀口知也, 東本崇仁, 平嶋宗: 知識モデリングに基づく学習支援, 人工知能学会誌, Vol. 35, No. 2, pp. 192–200 (2020)
- [今井 08] 今井功, 東本崇仁, 堀口知也, 平嶋宗: 中学理科における Error-based Simulation を用いた授業実践—「ニュートンに挑戦」プロジェクト—, 教育システム情報学会誌, Vol. 25, No. 2, pp. 194–203 (2008)
- [Inazawa 22] Inazawa, K. and Kashiwara, A.: Designing and Evaluating Presentation Avatar for Promoting Self-Review, *IEICE TRANSACTIONS on Information and Systems*, Vol. 105, pp. 1546–1556 (2022)

- [Ishino 18] Ishino, T., Goto, M., and Kashihara, A.: A robot for reconstructing presentation behavior in lecture, in *HAI 2018 - Proceedings of the 6th International Conference on Human-Agent Interaction*, pp. 67–75, ACM (2018)
- [Ishino 22] Ishino, T., Goto, M., and Kashihara, A.: Robot lecture for enhancing presentation in lecture, *Research and Practice in Technology Enhanced Learning*, Vol. 17, pp. 1–22 (2022)
- [岩井 21] 岩井健吾, 緒方祐, 林雄介, 平嶋宗: 異義可換性のある接続表現を対象とした接続詞演習環境の設計開発とその試験的利用, 電子情報通信学会論文誌 D, Vol. 104, No. 4, pp. 337–347 (2021)
- [Jimenez 15] Jimenez, F., Yoshikawa, T., Furuhashi, T., and Kanoh, M.: An Emotional Expression Model for Educational-Support Robots, *Journal of Artificial Intelligence and Soft Computing Research*, Vol. 5, pp. 51–57 (2015)
- [Jimenez 18] Jimenez, F., Yoshikawa, T., Furuhashi, T., and Kanoh, M.: Effects of a novel sympathy-expression method on collaborative learning among junior high school students and robots, *Journal of Robotics and Mechatronics*, Vol. 30, pp. 282–291 (2018)
- [神戸 15] 神戸健寛, 山元翔, 吉田祐太, 林雄介, 平嶋宗: 単文統合型作問学習支援システムの利用効果の問題構造把握の観点からの評価, 電子情報通信学会論文誌 D, Vol. 98, pp. 153–162 (2015)
- [Kamide 14] Kamide, H., Kawabe, K., Shigemi, S., and Arai, T.: Nonverbal behaviors toward an audience and a screen for a presentation by a humanoid robot, *Artificial Intelligence Research*, Vol. 3, pp. 57–66 (2014)
- [金森 14] 金森春樹, 東本崇仁, 米谷雄介, 赤倉貴子: プログラミングプロセスにおける「プログラムを読む学習」の提案及び「意味理解」プロセスの学習支援システムの開発, 電子情報通信学会論文誌 D, Vol. J97, No. 12, pp. 1843–1846 (2014)
- [神田 08] 神田崇行: コミュニケーションロボットによる学習支援, 人工知能, Vol. 23, pp. 229–236 (2008)
- [柏原 91] 柏原昭博, 平嶋宗, 中村祐一, 豊田順一: 対象の理解支援を目的とする ITS にお

- ける説明機能の高度化に関する検討-説明機能のためのモデル: EXSEL の提案, 電子情報通信学会論文誌 D, Vol. 74, pp. 1583–1595 (1991)
- [柏原 19] 柏原昭博: エンゲージメントを引き出す学習支援ロボット, コンピュータ&エデュケーション, Vol. 46, pp. 30–37 (2019)
- [柏原 20] 柏原昭博: ソーシャルロボットを用いた学びの研究, 教育システム情報学会誌, Vol. 37, pp. 73–82 (2020)
- [茅島 08] 茅島路子, 稲葉晶子, 溝口理一郎: メタ認知活動の困難さに関するフレームワークの提案, 教育システム情報学会誌, Vol. 25, pp. 19–31 (2008)
- [Kayashima 10] Kayashima, M. and Mizoguchi, R.: A new Perspective for Metacognition-driven Learning, in *Proceedings of the 18th International Conference on Computers in Education: ICCE 2010*, pp. 41–45 (2010)
- [來村 02a] 來村徳信, 笠井俊信, 吉川真理子, 高橋賢, 古崎晃司, 溝口理一郎: オントロジーに基づく機能的知識の体系的記述とその機能構造設計支援における利用, 人工知能学会論文誌, Vol. 17, pp. 73–84 (2002)
- [來村 02b] 來村徳信, 溝口理一郎: オントロジー工学に基づく機能的知識体系化の枠組み, 人工知能学会論文誌, Vol. 17, pp. 61–72 (2002)
- [北村 17] 北村拓也, 長谷浩也, 前田一誠, 林雄介, 平嶋宗: 論理構造の組み立て演習環境の設計開発と実験的評価, 人工知能学会論文誌, Vol. 32, No. 6, pp. C–H14_1–12 (2017)
- [Koedinger 07] Koedinger, K. R. and Alevan, V.: Exploring the assistance dilemma in experiments with cognitive tutors, *Educational Psychology Review*, Vol. 19, pp. 239–264 (2007)
- [古池 16] 古池謙人, 東本崇仁: 段階的な設計能力の向上を目的としたブロック育成型学習支援システムの提案, 人工知能学会第 78 回先進的学習科学と工学研究会, pp. 35–38 (2016)
- [古池 17] 古池謙人, 東本崇仁: プログラムにおける構造的理解のための部品の段階的拡張手法の提案とそのシステムの開発, 教育システム情報学会特集論文研究会研究報告=JSiSE research report, pp. 211–218, 教育システム情報学会 (2017)
- [古池 18a] 古池謙人, 東本崇仁: プログラミングにおける構造的理解のための部品の段階

- 的拡張手法の提案とそのシステムの開発, 教育システム情報学会誌, Vol. 35, No. 2, pp. 215–220 (2018)
- [Koike 18b] Koike, K., Tomoto, T., Horiguchi, T., and Hirashima, T.: Proposal of a Framework for a Stepwise Task Sequence in Programming, in Yamamoto, S. and Mori, H. eds., *HCI International 2018*, Vol. 10905 LNCS, pp. 266–277, Cham (2018), Springer International Publishing
- [Koike 18c] Koike, K., Tomoto, T., Horiguchi, T., and Hirashima, T.: Proposal of an adaptive programming-learning support system utilizing structuralized tasks, in *ICCE 2018 - 26th International Conference on Computers in Education, Workshop Proceedings*, pp. 278–287 (2018)
- [古池 19a] 古池謙人, 東本崇仁, 堀口知也, 平嶋宗: プログラミングの構造的理解を指向した部品の段階的拡張手法の提案と支援システムの開発・評価, 教育システム情報学会誌, Vol. 36, pp. 190–202 (2019)
- [Koike 19b] Koike, K., Tsuji, Y., Tomoto, T., Katagami, D., Obo, T., Ogai, Y., Sone, J., and Udagawa, Y.: Academic emotions affected by robot eye color: An investigation of manipulability and individual-adaptability, *International Journal of Advanced Computer Science and Applications*, Vol. 10, pp. 410–418 (2019)
- [古池 20] 古池謙人, 東本崇仁, 堀口知也, 平嶋宗: プログラミング学習における再利用性を指向した知識組織化のための知的支援—機能・振舞い・構造の観点に基づく問題解決過程のモデル—, 人工知能学会論文誌, Vol. 35, pp. C–J82_1–17 (2020)
- [古池 21] 古池謙人, 久世泰成, 東本崇仁: Error-based Simulation と概念マップを用いたメタ認知駆動学習を促すシステムの開発と評価, 電子情報通信学会論文誌 D, Vol. J104, No. 6, pp. 526–528 (2021)
- [古池 23a] 古池謙人, 樋村いづみ, 東本崇仁: TAME: 誤りの可視化を重視した Teachable Agent の挙動制御手法, 電子情報通信学会論文誌 D, Vol. J106-D, p. in press (2023)
- [古池 23b] 古池謙人, 藤島優希, 東本崇仁, 堀口知也, 平嶋宗: プログラミングの機能的階層構造の学習環境における習熟度を考慮した適応的フィードバック機能の評価, 教育システム情報学会誌, Vol. 40, p. in press (2023)

- [Kojima 12] Kojima, K., Muramatsu, K., and Matsui, T.: Experimental study toward estimation of a learner mental state from processes of solving multiple choice problems based on eye movements, in *Proceedings of the 20th International Conference on Computers in Education, ICCE 2012*, pp. 81–85 (2012)
- [Kölling 03] Kölling, M., Quig, B., Patterson, A., and Rosenberg, J.: The bluej system and its pedagogy, *International Journal of Phytoremediation*, Vol. 21, pp. 249–268 (2003)
- [倉山 12] 倉山めぐみ, 平嶋宗: 逆思考型を対象とした算数文章題の作問学習支援システムの設計開発と実践的利用, *人工知能学会論文誌*, Vol. 27, No. 2, pp. 82–91 (2012)
- [Le 13] Le, N.-T., Loll, F., and Pinkwart, N.: Operationalizing the continuum between well-defined and ill-defined problems for educational technology, *IEEE Transactions on Learning Technologies*, Vol. 6, pp. 258–270 (2013)
- [Leelawong 08] Leelawong, K. and Biswas, G.: Designing learning by teaching agents: The Betty’s Brain system, *International Journal of Artificial Intelligence in Education*, Vol. 18, No. 3, pp. 181–208 (2008)
- [Leite 14] Leite, I., Castellano, G., Pereira, A., Martinho, C., and Paiva, A.: Empathic robots for long-term interaction, *International Journal of Social Robotics*, Vol. 6, pp. 329–341 (2014)
- [Luxton-Reilly 18] Luxton-Reilly, A., Simon, , Albluwi, I., Becker, B. A., Gianakos, M., Kumar, A. N., Ott, L., Paterson, J., Scott, M. J., Sheard, J., and Szabo, C.: Introductory Programming: A Systematic Literature Review, in *Proceedings Companion of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education*, p. 55–106 (2018)
- [Lynch 09] Lynch, C., Ashley, K. D., Pinkwart, N., and Alevan, V.: Concepts, structures, and goals: Redefining ill-definedness, *International Journal of Artificial Intelligence in Education*, Vol. 19, pp. 253–266 (2009)
- [前田 15] 前田一誠, 平嶋宗, 小山正孝, 影山和也, 市村広樹: 児童の問題づくりを個別に促進する文章題作成コンピュータソフト及びカリキュラムの研究開発 (VII) : 乗除算

- 文章題の構造的理解を指向した作問学習支援システム設計・開発実践運用, 広島大学学部・附属学校共同研究紀要, Vol. 43, pp. 87–94 (2015)
- [Mark 93] Mark, M. A. and Greer, J. E.: Evaluation methodologies for intelligent tutoring systems, *Journal of artificial intelligence in education*, Vol. 4, pp. 129–153 (1993)
- [松田 97a] 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 D, Vol. 80, pp. 326–335 (1997)
- [松田 97b] 松田憲幸, 柏原昭博, 平嶋宗, 豊田順一: 再帰プログラミングを対象とした問題の比較に基づく問題機能の実現, 教育システム情報学会誌= Transactions of Japanese Society for Information and Systems in Education, Vol. 14, pp. 93–104 (1997)
- [Matsuda 13] Matsuda, N., Yarzebinski, E., Keiser, V., Raizada, R., Cohen, W. W., Stylianides, G. J., and Koedinger, K. R.: Cognitive anatomy of tutor learning: Lessons learned with SimStudent., *Journal of Educational Psychology*, Vol. 105, No. 4, pp. 1152–1163 (2013)
- [松田 18] 松田昇: 学習科学における教育エージェントの動向, 教育システム情報学会誌, Vol. 35, No. 1, pp. 13–20 (2018)
- [松居 10] 松居辰則, 平嶋宗: 学習課題・問題系列のデザイン, 人工知能学会誌, Vol. 25, No. 2, pp. 259–267 (2010)
- [Matsui 19] Matsui, T., Tawatsuji, Y., Fang, S., and Uno, T.: Conceptualization of IMS that estimates learners' mental states from learners' physiological information using deep neural network algorithm, in *International Conference on Intelligent Tutoring Systems*, pp. 63–71, Springer (2019)
- [松爲 22] 松爲泰生, 前田新太郎, 古池謙人, 東本崇仁: ソースコードの振る舞い制約と機能の理解を指向した学習支援環境の提案, 教育システム情報学会全国大会講演論文集, pp. 249–250 (2022)
- [松澤 14] 松澤芳昭, 保井元, 杉浦学, 酒井三四郎: ビジュアル-Java 相互変換によるシームレスな言語移行を指向したプログラミング学習環境の提案と評価, 情報処理学会論文誌, Vol. 55, pp. 57–71 (2014)

- [Milne 02] Milne, I. and Rowe, G.: Difficulties in learning and teaching programming - Views of students and tutors, *Education and Information Technologies*, Vol. 7, pp. 55–66 (2002)
- [溝口 95] 溝口理一郎：誤りを科学する：学習者モデルの構築, *人工知能学会誌*, Vol. 10, No. 3, pp. 348–353 (1995)
- [溝口 05] 溝口理一郎：オントロジー工学, オーム社 (2005)
- [森 11] 森 敏昭, 岡直樹, 中條和光：学習心理学: 理論と実践の統合をめざして, 培風館 (2011)
- [森元 07] 森元逞：C をさらに理解しながら学ぶデータ構造とアルゴリズム, 共立出版 (2007)
- [Muramatsu 16] Muramatsu, K., Tanaka, E., Watanuki, K., and Matsui, T.: Framework to describe constructs of academic emotions using ontological descriptions of statistical models, *Research and Practice in Technology Enhanced Learning*, Vol. 11, p. 5 (2016)
- [Nakahara 09] Nakahara, T., Konishi, T., Kogure, S., Noguchi, Y., and Itoh, Y.: Learning environment for algorithm and programming where learners operate objects in a domain world using GUI, *Proceedings of the 17th International Conference on Computers in Education, ICCE 2009*, Vol. 1, pp. 59–66 (2009)
- [中野 00] 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*, Vol. 83-D, pp. 539–549 (2000)
- [中野 02] 中野明, 平嶋宗, 竹内章：演算の理解を指向した作問学習支援システム, *人工知能学会論文誌*, Vol. 17, pp. 598–607 (2002)
- [西田 93] 西田豊明：定性推論の諸相, 朝倉書店 (1993)
- [西村 11] 西村多久磨, 河村茂雄, 櫻井茂男：自律的な学習動機づけとメタ認知的方略が学業成績を予測するプロセス, *教育心理学研究*, Vol. 59, pp. 77–87 (2011)
- [Nishimura 17] Nishimura, T. and Sakurai, S.: Longitudinal changes in academic motivation in Japan: Self-determination theory and East Asian cultures, *Journal of Applied Developmental Psychology*, Vol. 48, pp. 42–48 (2017)

- [Nkambou 10] Nkambou, R., Mizoguchi, R., and Bourdeau, J.: *Advances in intelligent tutoring systems*, Springer Science & Business Media (2010)
- [Nwana 90] Nwana, H. S.: Intelligent tutoring systems: an overview, *Artificial Intelligence Review*, Vol. 4, pp. 251–277 (1990)
- [Obaidallah 16] Obaidallah, U.: Comprehension and Composition of Flowcharts, in *PPIG 2016 - 27th Annual Workshop* (2016)
- [O’Brien 04] O’Brien, M. P., Buckley, J., and Shaft, T. M.: Expectation-based, inference-based, and bottom-up software comprehension, *Journal of Software Maintenance and Evolution*, Vol. 16, pp. 427–447 (2004)
- [荻田 22] 荻田将徳, 前田新太郎, 相川野々香, 古池謙人, 東本崇仁: 物理における制約の理解を指向した Teachable Agent Modeling for Error-visualization による学習手法の提案, 教育システム情報学会全国大会講演論文集, pp. 171–172 (2022)
- [Ohlsson 86] Ohlsson, S.: Some principles of intelligent tutoring, *Instructional science*, Vol. 14, pp. 293–326 (1986)
- [Okita 07] Okita, S. Y., Bailenson, J., and Schwartz, D. L.: The mere belief of social interaction improves learning, in *Proceedings of the annual meeting of the cognitive science society*, Vol. 29, pp. 1355–1360 (2007)
- [Okita 13] Okita, S. Y. and Schwartz, D. L.: Learning by teaching human pupils and teachable agents: The importance of recursive feedback, *Journal of the Learning Sciences*, Vol. 22, No. 3, pp. 375–412 (2013)
- [大槻 86] 大槻説乎: 高度個別教育における知識情報処理: ITS の機能とふるまい, 人工知能学会誌, Vol. 1, pp. 196–202 (1986)
- [大槻 00] 大槻説乎: 知的学習環境の構成論, 電子情報通信学会論文誌 D, Vol. 83, pp. 515–522 (2000)
- [Peitek 20] Peitek, N., Siegmund, J., Apel, S., Kästner, C., Parnin, C., Bethmann, A., Leich, T., Saake, G., and Brechmann, A.: A Look into Programmers’ Heads, *IEEE Transactions on Software Engineering*, Vol. 46, pp. 442–462 (2020)
- [Pekrun 02] Pekrun, R., Goetz, T., Titz, W., and Perry, R. P.: Academic emotions

- in students' self-regulated learning and achievement: A program of qualitative and quantitative research, *Educational Psychologist*, Vol. 37, pp. 91–105 (2002)
- [Pekrun 11] Pekrun, R., Goetz, T., Frenzel, A. C., Barchfeld, P., and Perry, R. P.: Measuring emotions in students' learning and performance: The Achievement Emotions Questionnaire (AEQ), *Contemporary Educational Psychology*, Vol. 36, pp. 36–48 (2011)
- [Pennington 87] Pennington, N.: Stimulus structures and mental representations in expert comprehension of computer programs, *Cognitive Psychology*, Vol. 19, pp. 295–341 (1987)
- [Quillian 68] Quillian, M. R.: Semantic memory, in Minsky, M. ed., *Semantic Information Processing*, MIT Press (1968)
- [Resnick 09] Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Y Silver, a J., Silverman, B., and Kafai, Y.: Scratch: Programming for All., *Communications of the ACM*, Vol. 52, pp. 60–67 (2009)
- [Rivers 17] Rivers, K. and Koedinger, K. R.: Data-driven hint generation in vast solution spaces: a self-improving python programming tutor, *International Journal of Artificial Intelligence in Education*, Vol. 27, pp. 37–64 (2017)
- [Ryan 89] Ryan, R. M. and Connell, J. P.: Perceived Locus of Causality and Internalization: Examining Reasons for Acting in Two Domains, *Journal of Personality and Social Psychology*, Vol. 57, pp. 749–761 (1989)
- [境 17a] 境 くりま, 港隆史, 石黒浩: わずかな感情変化を表現可能なアンドロイド動作の生成モデルの提案, 電子情報通信学会論文誌 D, Vol. 100, pp. 310–320 (2017)
- [Sakai 17b] Sakai, K., Minato, T., Ishi, C. T., and Ishiguro, H.: Novel speech motion generation by modeling dynamics of human speech production, *Frontiers in Robotics and AI*, Vol. 4, p. 49 (2017)
- [笹島 96] 笹島宗彦, 來村徳信, 池田満, 溝口理一郎: 機能と振舞いのオントロジーに基づく機能モデル表現言語 FBRL の開発, 人工知能学会誌, Vol. 11, pp. 420–431 (1996)

- [Schankin 18] Schankin, A., Berger, A., Holt, D. V., Hofmeister, J. C., Riedel, T., and Beigl, M.: Descriptive compound identifier names improve source code comprehension, in *Proceedings - International Conference on Software Engineering*, pp. 31–40, ACM (2018)
- [嶋川 21] 嶋川晋, 岩井健吾, 林雄介, 平嶋宗: 算数文章題の作問学習における段階的演習システムの設計・開発, 2020 年度 JSiSE 学生研究発表会, pp. 207–208 (2021)
- [新開 08] 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)
- [篠原 16] 篠原智哉, 今井功, 東本崇仁, 堀口知也, 山田敦士, 山元翔, 林雄介, 平嶋宗: 運動する物体にはたらく力を対象とした Error-based Simulation の中学校理科における実践利用, 電子情報通信学会論文誌 D, Vol. 99, No. 4, pp. 439–451 (2016)
- [篠原 17] 篠原智哉, 山田敦士, 林雄介, 平嶋宗: Error-Based Simulation による MIF 素朴概念の修正の効果の検証, 電子情報通信学会論文誌 D, Vol. 100, No. 3, pp. 447–450 (2017)
- [Shneiderman 77] Shneiderman, B.: Teaching programming: A spiral approach to syntax and semantics, *Computers & Education*, Vol. 1, No. 4, pp. 193–197 (1977)
- [Shneiderman 79] Shneiderman, B. and Mayer, R.: Syntactic/semantic interactions in programmer behavior: A model and experimental results, *International Journal of Parallel Programming*, Vol. 8, No. 3, pp. 219–238 (1979)
- [Sottolare 16] Sottolare, R. A., Graesser, A. C., Hu, X., Olney, A., Nye, B., and Sinatra, A. M.: *Design recommendations for intelligent tutoring systems: Volume 4 - domain modeling*, US Army Research Laboratory (2016)
- [杉浦 08] 杉浦学, 松澤芳昭, 岡田健, 大岩元: アルゴリズム構築能力育成の導入教育: 実作業による概念理解に基づくアルゴリズム構築体験とその効果, 情報処理学会論文誌, Vol. 49, pp. 3409–3427 (2008)
- [Supianto 16] Supianto, A. A., Hayashi, Y., and Hirashima, T.: Visualizations of problem-posing activity sequences toward modeling the thinking process, *Research and Practice in Technology Enhanced Learning*, Vol. 11, No. 1, p. 14 (2016)

- [Supianto 17a] Supianto, A. A., Hayashi, Y., and Hirashima, T.: An investigation of learner's actions in posing arithmetic word problem on an interactive learning environment, *IEICE Transactions on Information and Systems*, Vol. 100, No. 11, pp. 2725–2728 (2017)
- [Supianto 17b] Supianto, A. A., Hayashi, Y., and Hirashima, T.: Model-based analysis of thinking in problem posing as sentence integration focused on violation of the constraints, *Research and Practice in Technology Enhanced Learning*, Vol. 12, No. 1, pp. 1–21 (2017)
- [武智 15] 武智俊平, 林直也, 篠原智哉, 山元翔, 林雄介, 平嶋宗: 単純化方略を用いた問題解決失敗の自己克服支援システムとその実践的評価—初等力学を対象として—, *電子情報通信学会論文誌 D*, Vol. 98, No. 1, pp. 130–141 (2015)
- [竹中 18] 竹中真希子, 室田一成: 作問学習を取り扱った先行研究に関する基礎的研究: 先行研究で採用されている作問の方法, *大分大学教育学部研究紀要*, Vol. 40, pp. 133–148 (2018)
- [Tärning 19] Tärning, B., Silvervarg, A., Gulz, A., and Haake, M.: Instructing a Teachable Agent with Low or High Self-Efficacy – Does Similarity Attract?, *International Journal of Artificial Intelligence in Education*, Vol. 29, No. 1, pp. 89–121 (2019)
- [東本 08] 東本崇仁, 堀口知也, 平嶋宗: シミュレーションに基づく学習環境における漸進的な知識獲得支援のためのマイクロワールドグラフ, *電子情報通信学会論文誌 D*, Vol. 91, No. 2, pp. 303–313 (2008)
- [東本 16] 東本崇仁, 赤倉貴子: プログラムトレース課題の提案と学習支援システムの開発, *電子情報通信学会論文誌 D*, Vol. 99, No. 8, pp. 805–808 (2016)
- [東本 18] 東本崇仁, 赤倉貴子: 提案するプログラムトレース課題のための学習支援システムの開発とその実践, *電子情報通信学会論文誌 D*, Vol. 101, pp. 810–819 (2018)
- [Ueno 19] Ueno, U., Tomoto, T., Horiguchi, T., and Hirashima, T.: A support system for learning physics in which students identify errors using measurements displayed by a measurement tool, in *ICCE 2019 - 27th International Conference on Comput-*

- ers in Education, Proceedings*, Vol. 2, pp. 432–440 (2019)
- [植野 20] 植野和, 東本崇仁, 堀口知也, 平嶋宗: 学習者のパラメータ探索を通じた故障診断による力学学習支援システムの開発, 人工知能学会第 88 回先進的学習科学と工学研究会, pp. 103–108 (2020)
- [VanLehn 05] VanLehn, K., Lynch, C., Schulze, K., Shapiro, J. A., Shelby, R., Taylor, L., Treacy, D., Weinstein, A., and Wintersgill, M.: The Andes physics tutoring system: Lessons learned, *International Journal of Artificial Intelligence in Education*, Vol. 15, No. 3, pp. 147–204 (2005)
- [von Mayrhauser 95] von Mayrhauser, A. and Vans, A. M.: Program comprehension during software maintenance and evolution, *Computer*, Vol. 28, pp. 44–55 (1995)
- [渡辺 15] 渡辺圭祐, 東本崇仁, 赤倉貴子: 段階的抽象化を用いたプログラムを読む学習の支援システムの開発とその評価, 電子情報通信学会技術研究報告 = IEICE technical report : 信学技報, 第 115 巻, pp. 49–54, 電子情報通信学会 (2015)
- [Weiser 82] Weiser, M.: Programmers use slices when debugging, *Communications of the ACM*, Vol. 25, pp. 446–452 (1982)
- [Wenger 87] Wenger, E.: *Artificial intelligence and tutoring systems: computational and cognitive approaches to the communication of knowledge*, Morgan Kaufmann Publishers Inc. (1987)
- [Wing 06] Wing, J. M.: Computational thinking, *Communications of the ACM*, Vol. 49, No. 3, pp. 33–35 (2006)
- [Woolf 10] Woolf, B. P.: *Building intelligent interactive tutors: Student-centered strategies for revolutionizing e-learning*, Morgan Kaufmann (2010)
- [Xinogalos 15] Xinogalos, S.: Object-oriented design and programming: An investigation of novices' conceptions on objects and classes, *ACM Transactions on Computing Education*, Vol. 15, pp. 1–21 (2015)
- [山田 16] 山田敦士, 篠原智哉, 堀口知也, 林雄介, 平嶋宗: 多視点 Error-Based Simulation の設計・開発と実験的評価, 電子情報通信学会論文誌 D, Vol. 99, No. 12, pp. 1158–1161 (2016)

- [山本 14] 山本頼弥, 野口靖浩, 小暮悟, 山下浩一, 小西達裕, 伊東幸宏: 初学者に体系的デバッグ手法を学ばせるための学習環境, 教育システム情報学会, 第 39 回全国大会講演論文集, pp. 453–454 (2014)
- [山元 17] 山元翔, 赤尾優希, 室津光貴, 前田一誠, 林雄介, 平嶋宗: 算数文章題の構造的理解を指向した作問学習支援システムの乗除算への拡張とその実践利用, 電子情報通信学会論文誌 D, Vol. 100, No. 1, pp. 60–69 (2017)
- [山本 18] 山本頼弥, 野口靖浩, 小暮悟, 山下浩一, 小西達裕, 伊東幸宏: 場当たりのなデバッグを行ってしまう学習者に体系的デバッグ手順を指導する授業パッケージと学習支援システムの構築, 教育システム情報学会誌, Vol. 35, pp. 21–37 (2018)
- [Yamamoto 21] Yamamoto, S., Tobe, Y., Tawatsuji, Y., and Hirashima, T.: In-process Feedback by Detecting Deadlock based on EEG Data in Exercise of Learning by Problem-posing, in *ICCE 2021 - 29th International Conference on Computers in Education, Main Conference Proceedings*, pp. 21–30 (2021)
- [横山 06] 横山琢郎, 平嶋宗, 岡本真彦, 竹内章: 単文統合としての作問を対象とした学習支援システムの設計・開発, 教育システム情報学会誌, Vol. 23, pp. 166–175 (2006)
- [横山 07] 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, pp. 333–341 (2007)
- [Zapata-Rivera 04] Zapata-Rivera, J.-D. and Greer, J. E.: Interacting with in-spectable Bayesian student models, *International Journal of Artificial Intelligence in Education*, Vol. 14, pp. 127–163 (2004)

受賞一覧

1. 2021 年度 人工知能学会 先進的学習科学と工学 (ALST) 研究会 若手奨励賞, (2022.04) (相川野々香, **古池謙人**, 東本崇仁, 堀口知也, 平嶋宗: Error-based Simulation を対象にした因果推論理論に基づく自動的な補助問題生成法の検討, 人工知能学会第 92 回先進的学習科学と工学研究会, pp. 6–11, (2021))
2. 2021 年度 電子情報通信学会 教育工学研究会 研究奨励賞, (2021.12) (**古池謙人**, 樋村いづみ, 東本崇仁: 誤りの可視化を重視した Teachable Agent モデリング手法の提案, 電子情報通信学会技術研究報告 (ET), Vol. 121, No. 294, pp. 57–62, (2021))
3. 第 32 回教育・学習支援システム若手の会 最優秀奨励賞, (2021.10)
4. 2019 年度 東京工芸大学 杉浦六右衛門賞 受賞, (2020.03)
5. 2017 年度 東京工芸大学 工学部同窓会長賞 受賞, (2018.03)

業績一覧

原著論文（査読有）

- [α 1] 古池謙人, 樋村いづみ, 東本崇仁: TAME: 誤りの可視化を重視した Teachable Agent の挙動制御手法, 電子情報通信学会論文誌 D, Vol. J106-D, No. 2, in press, (2023)
- [α 2] 相川野々香, 古池謙人, 東本崇仁, 堀口知也, 平嶋宗: 力学を対象とした Error-based Simulation における行き詰まりの解消を指向した補助問題の提示システム, 電子情報通信学会論文誌 D, Vol. J106-D, No. 2, in press, (2023)
- [α 3] 相川野々香, 齊藤寛, 古池謙人, 東本崇仁: 力学における学習者の誤りに基づいた問題提示 (Error-based Problem Posing) システムの開発, 電子情報通信学会論文誌 D, Vol. J106-D, No. 2, in press, (2023)
- [α 4] 古池謙人, 東本崇仁, 堀口知也, 平嶋宗: プログラミング学習における再利用性を指向した知識組織化のための知的支援: 機能・振舞い・構造の観点に基づく問題解決過程のモデル, 人工知能学会論文誌, Vol. 35, No. 5, pp. C-J82_1-17, (2020)
- [α 5] 古池謙人, 東本崇仁, 堀口知也, 平嶋宗: プログラミングの構造的理解を指向した部品の段階的拡張手法の提案と支援システムの開発・評価, 教育システム情報学会誌, Vol. 36, No. 3, pp. 190–202, (2019)
- [α 6] Koike, K., Tsuji, Y., Tomoto, T., Katagami, D., Obo, T., Ogai, Y., Sone, J., and Udagawa, Y.: Academic Emotions Affected by Robot Eye Color: An Investigation of Manipulability and Individual-Adaptability, International Journal of Advanced Computer Science and Applications, Vol. 10, No. 4, pp. 410–418, (2019)

レター（査読有）

- [β 1] 前田新太郎, 茂木誠拓, 古池謙人, 東本崇仁: 仮想ロボットプログラミングを用いたコード共有プラットフォームの開発と評価, 教育システム情報学会誌, Vol. 40, No.

- 3, in press, (2023)
- [β2] 古池謙人, 藤島優希, 東本崇仁, 堀口知也, 平嶋宗: プログラミングの機能的階層構造の学習環境における習熟度を考慮した適応的フィードバック機能の評価, 教育システム情報学会誌, Vol. 40, No. 3, in press, (2023)
- [β3] 古池謙人, 久世泰成, 東本崇仁: Error-based Simulation と概念マップを用いたメタ認知駆動学習を促すシステムの開発と評価, 電子情報通信学会論文誌 D, Vol. J104-D, No. 6, pp. 526–528, (2021)
- [β4] 相川野々香, 古池謙人, 東本崇仁: Error-based Simulation (EBS) における学習者の誤り傾向に基づく問題の行詰まり解決支援システム, 電子情報通信学会論文誌 D, Vol. J103-D, No. 9, pp. 644–647, (2020)
- [β5] 古池謙人, 東本崇仁: プログラミングにおける構造的理解のための部品の段階的拡張手法の提案とそのシステムの開発, 教育システム情報学会誌, Vol. 35, No. 2, pp. 215–220, (2018)

国際会議（査読有）

- [γ1] Aikawa, N., Maeda, S., Mogi, T., **Koike, K.**, Tomoto, T., Imai, I., Horiguchi, T., and Hirashima, T.: Practical Use of an Error-based Problem Presentation System in Mechanics, Proceedings of the International Conference on Computers in Education ICCE 2022, pp. 118–123, (2022)
- [γ2] **Koike, K.**, Himura, I., and Tomoto, T.: TAME: A Method of Teachable Agent Modeling for Error-Visualization, HCI International 2022, Vol. 13305 LNCS, pp. 461–474, (2022)
- [γ3] Aikawa, N., **Koike, K.**, Tomoto, T., Horiguchi, T., and Hirashima, T.: Automatic Generation Rules for Auxiliary Problems Based on Causal Relationships for Force in a Mechanics Learning Support System, HCI International 2022, Vol. 13305 LNCS, pp. 437–450, (2022)
- [γ4] Maeda, S., **Koike, K.**, and Tomoto, T.: A Knowledge Sharing Platform for

- Learning from Others' Code, HCI International 2022, Vol. 13305 LNCS, pp. 489–504, (2022)
- [γ5] Aikawa, N., **Koike, K.**, Tomoto, T., Horiguchi, T., and Hirashima, T.: Characterization of Auxiliary Problems for Automated Generation in Error-based Simulation, HCI International 2021, Vol. 12766 LNCS, pp. 3–13, (2021)
- [γ6] **Koike, K.**, Fujishima, Y., Tomoto, T., Horiguchi, T., and Hirashima, T.: Learner Model for Adaptive Scaffolding in Intelligent Tutoring Systems for Organizing Programming Knowledge, HCI International 2021, Vol. 12766 LNCS, pp. 63–74, (2021)
- [γ7] Mogi, T., **Koike, K.**, Tomoto, T., Horiguchi, T., and Hirashima, T.: Proposal of Learning Support System for Improving Skills in Inferring Background Knowledge in Conversation, HCI International 2021, Vol. 12766 LNCS, pp. 104–114, (2021)
- [γ8] **Koike, K.**, Mogi, T., Tomoto, T., Horiguchi, T., and Hirashima, T.: Com-pogram: Development and Evaluation of ITS for Organizing Programming-Knowledge by Visualizing Behavior, HCI International 2020, Vol. 12427 LNCS, pp. 151–162, (2020)
- [γ9] **Koike, K.**, Tomoto, T., Horiguchi, T., and Hirashima, T.: Proposal of a Framework for a Stepwise Task Sequence in Programming, HCI International 2018, Vol. 10905 LNCS, pp. 266–277, (2018)

ワークショップ (査読有)

- [δ1] Aikawa, N., **Koike, K.**, and Tomoto, T.: Analysis of Learning Activities with Automated Auxiliary Problem Presentation for Breaking Learner Impasses in Physics Error-based Simulations, Workshop proceedings of the International Conference on Computers in Education ICCE 2020, pp. 72–83, (2020)
- [δ2] **Koike, K.**, Tomoto, T., Horiguchi, T., and Hirashima, T.: Supporting Knowl-

- edge Organization for Reuse in Programming: Proposal of a System Based on Function—Behavior—Structure Models, Workshop proceedings of the International Conference on Computers in Education ICCE 2019, pp. 382–392, (2019)
- [δ3] **Koike, K.**, Tomoto, T., Horiguchi, T., and Hirashima, T.: Proposal of an Adaptive Programming-Learning Support System Utilizing Structuralized Tasks, Workshop proceedings of the International Conference on Computers in Education ICCE 2018, pp. 278–287, (2018)
- [δ4] **Koike, K.**, Tomoto, T., and Hirashima, T.: Proposal of a Stepwise Support for Structural Understanding in Programming, Workshop proceedings of the International Conference on Computers in Education ICCE 2017, pp. 471–481, (2017) (Oral Presentation and Interactive Session)

研究会

- [ε1] 相川野々香, 前田新太郎, 茂木誠拓, **古池謙人**, 東本崇仁, 今井功, 堀口知也, 平嶋宗: 物理学を対象とした誤りに基づく問題提示システムを用いた授業実践におけるテスト結果の分析, 電子情報通信学会技術研究報告 (ET), Vol. 122, No. 241, pp. 11-18, (2022)
- [ε2] 前田新太郎, **古池謙人**, 東本崇仁: 洗練活動を促すコード共有プラットフォームにおける戦略を基にしたコードの共有手法, 電子情報通信学会技術研究報告 (ET), Vol. 122, No. 241, pp. 25-28, (2022)
- [ε3] 松為泰生, 前田新太郎, **古池謙人**, 東本崇仁: ソースコードの振舞いモデルから機能の解釈を促す支援システムにおけるフィードバックの検討, 電子情報通信学会技術研究報告 (ET), Vol. 122, No. 241, pp. 29-34, (2022)
- [ε4] 松為泰生, 前田新太郎, **古池謙人**, 東本崇仁: ソースコードの振舞い制約の思考から機能の解釈を促す支援システムの提案, 人工知能学会第 95 回先進的学習科学と工学研究会, pp. 11–14, (2022)
- [ε5] 前田新太郎, **古池謙人**, 東本崇仁: 仮想ロボットプログラミングを対象としたコード

- の洗練活動を促す知識共有プラットフォームの実践利用の分析, 人工知能学会第 95 回先進的学習科学と工学研究会, pp. 1–6, (2022)
- [ε6] 松為泰生, 前田新太郎, 茂木誠拓, **古池謙人**, 東本崇仁: 振る舞いのトレースを通じてソースコードの機能の理解を促す学習支援システムの開発, 人工知能学会第 94 回先進的学習科学と工学研究会, pp. 13–18, (2022)
- [ε7] 齊藤寛, 相川野々香, **古池謙人**, 東本崇仁: 力学における学習者の誤答に基づいた誤り作問システムの開発, 人工知能学会第 94 回先進的学習科学と工学研究会, pp. 7–12, (2022)
- [ε8] 石井翔大, **古池謙人**, 東本崇仁: 人狼ゲームにおける振り返り活動を通じた自責的思考を促す支援システムの開発と評価, 電子情報通信学会技術研究報告 (ET), Vol. 121, No. 406, pp. 123–128, (2022)
- [ε9] 相川野々香, 前田新太郎, 茂木誠拓, **古池謙人**, 東本崇仁, 今井功, 堀口知也, 平嶋宗: Error-based Simulation における学習者の誤り箇所に適応的な補助問題提示システムの中学校での授業実践, 2021 年度 JSiSE 学生研究発表会, pp. 67–68, (2022)
- [ε10] 松為泰生, 前田新太郎, 茂木誠拓, **古池謙人**, 東本崇仁: プログラミングにおける振る舞いと機能の理解を指向した学習支援システム, 2021 年度 JSiSE 学生研究発表会, pp. 41–42, (2022)
- [ε11] **古池謙人**, 樋村いづみ, 東本崇仁: 誤りの可視化を重視した Teachable Agent モデリング手法の提案, 電子情報通信学会技術研究報告 (ET), Vol. 121, No. 294, pp. 57–62, (2021)
- [ε12] 相川野々香, **古池謙人**, 東本崇仁, 堀口知也, 平嶋宗: Error-based Simulation を対象にした因果推論理論に基づく自動的な補助問題生成法の検討, 人工知能学会第 92 回先進的学習科学と工学研究会, pp. 6–11, (2021)
- [ε13] 前田新太郎, **古池謙人**, 東本崇仁: プログラミングを対象とした競争型知識共有プラットフォームにおける評価指標の妥当性検証, 人工知能学会第 92 回先進的学習科学と工学研究会, pp. 12–17, (2021)
- [ε14] 茂木誠拓, **古池謙人**, 東本崇仁, 堀口知也, 平嶋宗: すれ違いの解消を指向した背景知識を推定する能力を育成する学習支援システムの開発, 人工知能学会第 92 回先進的

- 学習科学と工学研究会, pp. 31–36, (2021)
- [€15] 太田光一, 油谷知岐, **古池謙人**, 堀越泉, 田和辻可昌, 山元翔: 第 31 回 教育・学習支援システム若手の会 報告, 人工知能学会第 91 回先進的学習科学と工学研究会, pp. 28–33, (2021)
- [€16] 座間出実, **古池謙人**, 東本崇仁: 部品の利用と組み合わせを指向したプログラミング学習支援システムの開発と評価, 人工知能学会第 91 回先進的学習科学と工学研究会, pp. 54–59, (2021)
- [€17] 藤島優希, **古池謙人**, 東本崇仁: 知識の再利用性向上を目的としたプログラミング学習における理解状態に基づく適応的フィードバックの開発と評価, 人工知能学会第 91 回先進的学習科学と工学研究会, pp. 60–65, (2021)
- [€18] 樋村いづみ, **古池謙人**, 東本崇仁: 算数文章題の制約理解を目的とした教育エージェントを伴う作問学習支援システムの開発, 人工知能学会第 91 回先進的学習科学と工学研究会, pp. 79–86, (2021)
- [€19] 前田新太郎, **古池謙人**, 東本崇仁: ロボットプログラミングを題材にした競争型知識共有プラットフォームの提案と実装, 人工知能学会第 91 回先進的学習科学と工学研究会, pp. 87–92, (2021)
- [€20] 茂木誠拓, **古池謙人**, 東本崇仁, 堀口知也, 平嶋宗: すれ違いの解消スキル育成を指向した学習支援システムの提案, 人工知能学会第 91 回先進的学習科学と工学研究会, pp. 99–104, (2021)
- [€21] 相川野々香, **古池謙人**, 東本崇仁, 堀口知也, 平嶋宗: Error-based Simulation における自動的な補助問題生成法の提案, 人工知能学会第 91 回先進的学習科学と工学研究会, pp. 105–110, (2021)
- [€22] 前田新太郎, **古池謙人**, 東本崇仁: 競争型知識共有プラットフォームを用いたロボットプログラミングによる学習支援システムの検討, 2020 年度 JSiSE 学生研究発表会, pp. 51–52, (2021)
- [€23] 茂木誠拓, **古池謙人**, 東本崇仁, 堀口知也, 平嶋宗: すれ違いの解消に向けた背景知識の推論能力を育成する学習支援システムの提案, 2020 年度 JSiSE 学生研究発表会, pp. 85–86, (2021)

- [€24] 相川野々香, **古池謙人**, 東本崇仁: 行詰り解決を指向した Error-based Simulation システムの補助問題提示機能による学習活動の分析, 人工知能学会第 89 回先進的学習科学と工学研究会, pp. 17-20, (2020)
- [€25] 座間出実, **古池謙人**, 東本崇仁: プログラミングにおける部品の利用と合成を指向したプログラミング学習支援システムの開発, 人工知能学会第 89 回先進的学習科学と工学研究会, pp. 21-24, (2020)
- [€26] 茂木誠拓, **古池謙人**, 東本崇仁: 技術仕様書における処理と対象に焦点を当てた文章読解の能力を育成する学習支援環境の検討, 人工知能学会第 89 回先進的学習科学と工学研究会, pp. 25-28, (2020)
- [€27] 茂木誠拓, **古池謙人**, 東本崇仁: 要求分析能力の獲得支援を指向した学習環境に対するフィードバック機能による効果の検証, 人工知能学会第 88 回先進的学習科学と工学研究会, pp. 61-64, (2020)
- [€28] 茂木誠拓, **古池謙人**, 東本崇仁: 要求分析における処理の明確化を指向した学習環境における学びの検証と行き詰まり箇所の考察, 人工知能学会第 87 回先進的学習科学と工学研究会, pp. 1-6, (2019)
- [€29] **古池謙人**, 東本崇仁, 堀口知也, 平嶋宗: プログラミングにおける学習者の部品構築プロセスを考慮した学習支援システムの提案, 人工知能学会第 86 回先進的学習科学と工学研究会, pp. 66-69, (2019)
- [€30] 茂木誠拓, **古池謙人**, 東本崇仁: システム開発における要求具体化タスクと支援環境の提案, 人工知能学会第 86 回先進的学習科学と工学研究会, pp. 89-92, (2019)
- [€31] **古池謙人**, 東本崇仁, 堀口知也, 平嶋宗: プログラミングにおける部品の獲得・拡張活動プロセス, 人工知能学会第 85 回先進的学習科学と工学研究会, pp. 104-109, (2019)
- [€32] **古池謙人**, 東本崇仁, 堀口知也, 平嶋宗: プログラミング課題を構造化することによる理解状態に適応した課題提示方法の検討, 人工知能学会第 83 回先進的学習科学と工学研究会, pp. 35-40, (2018)
- [€33] 辻優也, **古池謙人**, 東本崇仁: ロボットによる教授行動に用いた色情報が人の印象に与える影響, 教育システム情報学会第 1 回研究会研究報告, Vol. 33, No. 1, pp. 1-8,

(2018)

- [ε34] 古池謙人, 東本崇仁, 堀口知也, 平嶋宗: プログラミング学習を対象とした課題系列の拡張, 人工知能学会第 82 回先進的学習科学と工学研究会, pp. 7–12, (2018)
- [ε35] 古池謙人, 東本崇仁, 堀口知也, 平嶋宗: プログラミング課題を対象とした課題系列の構築, 2017 年度 JSiSE 学生研究発表会, p. A04, (2018)
- [ε36] 古池謙人, 東本崇仁, 堀口知也, 平嶋宗: プログラミングにおける段階的な課題系列の検討, 電子情報通信学会技術研究報告 (ET), Vol. 117, No. 296, pp. 47–52, (2017)
- [ε37] 古池謙人, 東本崇仁: プログラムにおける構造的理解のための部品の段階的拡張手法の提案とそのシステムの開発, 教育システム情報学会特集論文研究会研究報告=JSiSE research report, Vol. 8, No. 8, pp. 211–218, (2017)
- [ε38] 古池謙人, 東本崇仁: プログラミング学習における部品の段階的拡張手法の提案, 2017 年度 JSiSE 学生研究発表会, p. B01, (2017)
- [ε39] 綾部沙樹, 古池謙人, 東本崇仁: 複数科目の階層化に基づいた学習方法の転移支援システムの開発, 人工知能学会第 79 回先進的学習科学と工学研究会, pp. 45–50, (2017)
- [ε40] 古池謙人, 東本崇仁: 段階的な設計能力の向上を目的としたブロック育成型学習支援システムの提案, 人工知能学会第 78 回先進的学習科学と工学研究会, pp. 35–38, (2016)
- [ε41] 古池謙人, 東本崇仁: FPS を対象とした瞬間判断能力の育成システム, 人工知能学会第 75 回先進的学習科学と工学研究会, pp. 50–53, (2015)

全国大会

- [ζ1] 荻田将徳, 相川野々香, 古池謙人, 東本崇仁: 物理を題材とした Teachable Agent Modeling for Error-visualization における制約の提案, 第 24 回日本感性工学会大会, p. [1D-1-07], (2022)
- [ζ2] 松爲泰生, 前田新太郎, 古池謙人, 東本崇仁: ソースコードの機能の理解を振舞いモデルの解釈から促す支援システムの提案, 第 24 回日本感性工学会大会, p. [1D-1-06], (2022)

- [ζ3] 荻田将徳, 前田新太郎, 相川野々香, **古池謙人**, 東本崇仁: 物理における制約の理解を指向した Teachable Agent Modeling for Error-visualization による学習手法の提案, 教育システム情報学会全国大会講演論文集, pp. 171–172, (2022)
- [ζ4] 前田新太郎, **古池謙人**, 東本崇仁: より良いコードへ洗練する活動を促す仮想ロボットプログラミングを用いたコード共有プラットフォームの授業実践, 教育システム情報学会全国大会講演論文集, pp. 209–210, (2022)
- [ζ5] 松爲泰生, 前田新太郎, **古池謙人**, 東本崇仁: ソースコードの振る舞い制約と機能の理解を指向した学習支援環境の提案, 教育システム情報学会全国大会講演論文集, pp. 249–250, (2022)
- [ζ6] 相川野々香, 前田新太郎, 茂木誠拓, **古池謙人**, 東本崇仁, 今井功, 堀口知也, 平嶋宗: 力学における学習者の誤りに適応的な問題提示システムの授業実践の報告, 教育システム情報学会全国大会講演論文集, pp. 301–302, (2022)
- [ζ7] 相川野々香, **古池謙人**, 東本崇仁: 力学学習支援システムにおける補助問題の自動生成ルールの提案, 第 23 回日本感性工学会大会, p. [1D01-10-03], (2021)
- [ζ8] 相川野々香, **古池謙人**, 東本崇仁: Error-based Simulation システムにおける行詰り解決のための学習履歴を用いた補助問題提示方法の提案, 第 22 回日本感性工学会大会, p. [2B06-10-04], (2020)
- [ζ9] **古池謙人**, 茂木誠拓, 東本崇仁, 堀口知也, 平嶋宗: 振舞いを可視化することによるプログラミング学習支援システムの開発, 第 22 回日本感性工学会大会, p. [2B06-10-05], (2020)
- [ζ10] Mogi, T., **Koike, K.**, and Tomoto, T.: Evaluation of feedback in a learning environment to improve requirement analysis skills, The 34th Annual Conference of the Japanese Society for Artificial Intelligence, p. [1G4-ES-5-05], (2020)
- [ζ11] **古池謙人**, 中村遥佳, 遠藤杏奈, 油谷知岐: 志学の会: 2019 年度 年間活動報告 - 感性工学における入り口としての在り方と展望 -, 日本感性工学会春季大会, p. [1E-01], (2020)
- [ζ12] 酒川拓也, **古池謙人**, 東本崇仁: 連続的なインタラクションにおいてロボットの発色を伴う教授行動が学習者の心的状態に与える影響, 日本感性工学会春季大会, p.

- [1E-04], (2020)
- [ζ13] 古池謙人, 東本崇仁, 堀口知也, 平嶋宗: プログラミング学習での問題解決過程を考慮する学習支援システムの試作, 教育システム情報学会全国大会講演論文集, pp. 447–448, (2019)
- [ζ14] 酒川拓也, 古池謙人, 東本崇仁: ロボットの教授行動を用いて学習者の心的状態をコントロールする手法の検討, 教育システム情報学会全国大会講演論文集, pp. 405–406, (2019)
- [ζ15] 茂木誠拓, 古池謙人, 東本崇仁: システム開発における要求具体化スキル獲得支援環境の構築, 教育システム情報学会全国大会講演論文集, pp. 99–100, (2019)
- [ζ16] 古池謙人, 東本崇仁, 堀口知也, 平嶋宗: プログラミングにおける構造的理解のための部品の段階的拡張手法を用いたシステムの評価, 教育システム情報学会全国大会講演論文集, pp. 409–410, (2018)
- [ζ17] 辻優也, 古池謙人, 東本崇仁: ロボットによる学習者に適応的な色情報を活用した教授行動の検討, 教育システム情報学会全国大会講演論文集, pp. 313–314, (2018)
- [ζ18] 古池謙人, 東本崇仁: ボトムアップ型アプローチを用いたプログラミング学習手法の提案, 教育システム情報学会全国大会講演論文集, pp. 255–256, (2017)
- [ζ19] 古池謙人, 山添崇, 東本崇仁, 東吉彦: ヒトの主観的印象に基づくロゴデザインの自動評価システムの開発, 情報処理学会第 78 回全国大会講演論文集, pp. 715–716, (2016)

寄稿

- [η1] 古池謙人, 布施陽太郎: 学生企画「観点に基づく研究のデザイン～主観から客観へ～」, 人工知能学会誌, Vol. 36, No. 6, p. 744, (2021)
- [η2] 油谷知岐, 古池謙人: 学生企画「各世代の AI 研究者に問う — 『善く生きる』の捉え方—」, 人工知能学会誌, Vol. 35, No. 6, pp. 803–804, (2020)

その他

- [01] 古池謙人, 東本崇仁, 堀口知也, 平嶋宗: プログラミング的思考を促すための段階的
理解手法, 日本感性工学会春季大会, WP2-17, (2018)
- [02] 古池謙人, 山添崇, 東吉彦, 東本崇仁: 視覚刺激の提示位置が一時記憶に及ぼす影響,
第 8 回 Ambient Feedback System 研究会×日本感性工学会而立の会 2017 年度研
究会, (2017)