

認知モデル構築を経験することによる学習の予備的検討

Preliminary Study on Learning by Experiencing Construction of a Cognitive Model

小島一晃^{1*} 三輪和久² 中池竜一³ 神崎奈奈⁴
寺井仁⁵ 森田純哉⁶ 齋藤ひとみ⁷ 松室美紀²
Kazuaki Kojima¹ Kazuhisa Miwa² Ryuichi Nakaike³ Nana Kanzaki⁴
Hitoshi Terai⁵ Jun'ya Morita⁶ Hitomi Saito⁷ Miki Matsumuro²

¹ 帝京大学 ² 名古屋大学 ³ 平安女学院大学
¹ Teikyo University ² Nagoya University ³ Heian Jogakuin University
⁴ 名古屋女子大学 ⁵ 近畿大学
⁴ College of Nagoya Women's University ⁵ Kindai University
⁶ 静岡大学 ⁷ 愛知教育大学
⁶ Shizuoka University ⁷ Aichi University of Education

Abstract: Construction of models is promising as a learning activity. However, it can be a difficult activity which imposes heavy load on learners because it requires eminent skills. This study designed a basic framework for learning by experiencing construction of models on a production system in the domain of cognitive science. In this framework, a model abstractly describing human thinking processes and its computer model implemented on the production system is prepared by an instructor in advance. A learner is given the abstract model and thinking processes produced by executing the implementation model, and then engaged in instantiating the abstract model into an implementation model. This activity is expected to deepen learner understanding of thinking embedded in the abstract model. We preliminarily studied the effect of learning a model which simulates subtraction requiring regrouping in the framework. The results indicated the possibility that such experience can improve learner understanding of thinking behind the model, and necessity to expand learning activities in the framework.

1 はじめに

過去数十年の科学研究は、対象を理解する方法として対象の観察を通じた分析的アプローチと、対象のモデルの構築とシミュレーションを通じた構成的アプローチの2つを主に使用してきた。例えば認知科学の領域では、人間の心を理解するために、実験を通じて人間の行動を経験的に検討する方法の他に、計算機モデルを実行する方法を採用してきた [8]。

モデルを作ることによる対象理解は、一般教育における学習活動としても有望であると考えられる。しかし、プログラミング言語を用いてモデルを実装するといったような、高度なスキルを要求する活動でもあるため、多くの学習者にとって負荷が高く困難であると

いう問題がある。ここでは、モデルを作ることは手段であって学習目標ではないため、プログラミングのスキル育成を目的とする教育課程以外では、このスキルを訓練することなく学習活動を成立させることが必要となる。モデルを作る学習活動の支援の実現については、既に研究例がある（例えば [1, 2]）。これらの支援を受ける学習者は、対象に含まれるエージェントやオブジェクトの属性・振る舞いなどを抽象的に記述することで、モデルを作成できる。そして、計算機が実行可能なモデルへの具体化は支援システムによって代行され、モデルの実行を通じて学習者の誤概念などへの気づきが促進される。本研究では、前者のような抽象度の高い記述を「抽象モデル」、後者の具体化されたものを「実装モデル」と呼ぶことにする。これらの支援は、抽象モデルのみでの学習を可能にすると言うことができよう。

モデル構築においては、対象理解を外化したものが

*連絡先：帝京大学ラーニングテクノロジー開発室
栃木県宇都宮市豊郷台 1-1
E-mail: kojima@lt-lab.teikyo-u.ac.jp

抽象モデルであり、これを作成することが本質的である。一方で、認知科学の歴史に示されるように、抽象モデルの記述のみに留まらず実装モデルも作成することが、対象理解を深める上で重要な役割を果たすと考えられる。モデル構築による対象理解においては、抽象モデルから実装モデルへと具体化がなされることで実世界または仮想世界から受け取るフィードバックが、重要な要素のひとつである [5, 7]。また、人間の問題解決における思考過程などは、エージェントやオブジェクトとそのインタラクションとしてモデルを表現することが難しいと考えられる。このような領域でモデル構築による学習を支援するためには、異なるアプローチが必要である。

本研究では、認知科学の領域における人間の思考過程を対象として、モデルを作ることによる学習のフレームワークを設計した。このフレームワークでは、学習者には抽象モデルが与えられ、これを実装モデルに具体化することを、比較的低い負荷で経験することが可能である。そして、このフレームワークにおいて学習が成立し得るかを、実験を通じて予備的に検討した。

2 認知モデル構築を経験することによる学習のフレームワーク

本研究では学習対象を人間の思考過程とし、実装モデルのアーキテクチャとしてプロダクションシステムを採用する。そして、プロダクションモデルの構築を経験することで、人間の認知について学ぶフレームワークを提案する。

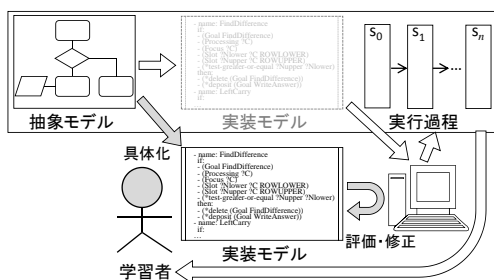


図 1: フレームワークの概念図

このフレームワークの概念図を図 1 に示す。概念モデルと実装モデルは、教授者によってあらかじめ用意される。学習者には概念モデルと、実装モデルから得られる実行過程が与えられる。そして、この実行過程を参考に、概念モデルを実行モデルへと具体化する。我々は先行研究において、対象を再産出する活動が、対象の構造に対する理解を促進することを確認している [4]。そのため、このフレームワークによる学習活動では、概念モデルに埋め込まれた現象に対する理解を深めることが期待される。

3 支援システムの試作

本研究では、前節で述べたフレームワークにおける学習を支援するシステムを試作した。本システムでは、初学者用に設計されたプロダクションシステムである DoCoPro [6] を、実装モデルのアーキテクチャとして使用する。システムを学習者に与える前に、教授者は概念モデルに対する実装モデルを DoCoPro 上で作成する。システムは実装モデルを実行し、図 2 に示される構造を持つモデル実行過程の情報を作成する。これには、モデルの実行過程における各ステップで発火したプロダクションルールの説明¹と、ルール発火前・後のワーキングメモリの状態（それぞれ前状態、後状態と呼ぶ）が含まれる。

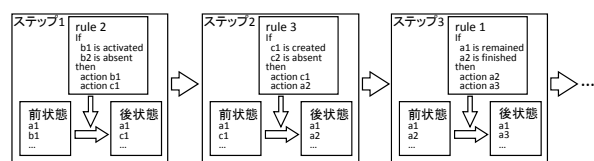


図 2: モデル実行過程の構造

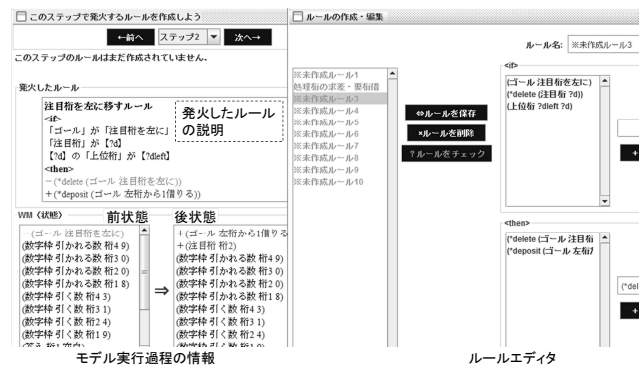


図 3: システムの画面の一部

学習者は概念モデルを与えられた後、システム上でその具体化を行う。システムの画面を図 3 に示す。図の左には、モデル実行過程の各ステップにおけるルールと前状態、後状態が提示される。学習者はこれを参照しながら、前状態を後状態に書き換えるルールを図の右に示されるエディタ上で作成する。作成したルールは、それを発火することでどのような状態になるか、後状態と比較して違いがあるかのフィードバックを受け取る形でチェックする。これは、モデルの実行過程をトレースしながらルールを作成する活動と言えるが、ルールを構成する情報はほぼ与えられている形になっており、ルールの作成を極めて容易にしている。これにより、プログラミングのスキルを持たない学習者であっても、概念モデルの具体化と、そこからのフィードバックを経験することが可能となる。

¹ルールに含まれる述語を、テンプレートを用いてより自然言語的な記述に変換したもので構成される

4 予備実験

前節の試作システムを用いてモデル作成を経験することで学習が起こるかを、経験的に確認した。モデルの題材には、先行研究 [3] で使用された、繰り下がりのある引き算の筆算を使用した。これは、誰もが実行可能であるが、手続き化され無意識に実行される思考である。そのため、モデル作成を通じて無意識化された思考過程に気づき、理解を深める題材として適していると考えられる。なお、先行研究では認知科学の授業実践において DoCoPro を使用し、プロダクションシステムの導入を 1 コマで実施した後、モデルの作成について解説したテキストとモデル実行過程における各ステップの状態を視覚化する機能を用いて、実装モデルの作成を 2 コマで、さらに誤った手続きにより誤答を起こすバグモデルの作成を 1 コマで実施している。

4.1 方法

本実験の参加者は、プログラミングに特に熟達していない一般大学生 8 名であった。参加者は実験の前に、先行研究 [3] の導入で使用された題材を用いて、プロダクションシステムについて各自で学習した。そして実験では、まず事前テストに回答した。事前テストでは、引き算の筆算 1 題 (317-98) への解答、引き算の手続き説明課題への回答、2 つの引き算の誤回答 (9008-3149=5959 と 806303-182465=623938) の原因を推定する課題 (バグ同定課題) への回答が求められた。筆算への解答は正しく計算できるかを試すためではなく、手続き説明課題の前に繰り下がりの処理の手続きを意識させるために設定したものである。続いて、システムの操作方法が動画教材によって説明された。そして、引き算の筆算の概念モデルと、実装モデルに使用された問題表現の解説が提示され、システムを用いて実装モデルへと具体化した。その後、事後テストにおいて、手続き説明課題とバグ同定課題への再度の回答と、実装モデルの具体化において学んだことや気付いたことの報告が求められた。

手続き説明課題の分析では、モデルに含まれる 10 のルールに相当する記述があるかどうかの評価された。相当する記述が含まれる場合は「あり」、記述があるが条件や操作が特殊化されていたり不十分である場合は「不完全」、含まれない場合は「なし」とした。10 のルールは下記である。

FindDifference1 これから計算する桁 (処理桁) の引かれる数が引く数以上の時 (WriteAnswer に移る)

WriteAnswer 処理桁の引かれる数から引く数の差を書く

ShiftColumn 処理桁を 1 つ左の桁に移す

Completed 最も左の桁の差を書き終えたら終了する

FindDifference2 処理桁の引かれる数が引く数より小さい時 (LeftCarry に移る)

LeftCarry 数を借りる桁 (注目桁) を 1 つ左に移す

GetCarry1 注目桁の引かれる数が 0 でなければ、注目桁を 1 減じて注目桁を 1 つ右に移す

PutCarry1 注目桁に 10 を加える (FindDifference に移る)

GetCarry2 注目桁の引かれる数が 0 であれば (LeftCarry に移る)

PutCarry2 注目桁に 10 を加える (GetCarry に移る)

たとえば PutCarry2 は、操作の記述が「1 の位に 10 を足す」や「1 つ右の位に 10 を足す」であれば「不完全」となる。

バグ同定課題は先行研究で使用された課題であるため、これと同様に 2 つの誤答の原因を正しく記述したら 2 点、2 つの誤答を一貫したルールで説明しているが誤っていたら 1 点、現象の記述をしているのみか 2 つの誤答を異なるルールで説明していたら 0 点とした。

4.2 結果

事前テストの筆算への解答は、全員が正解した。事前・事後テストの手続き説明課題における各ルールの評価を図 4 に示す。図に示されるように、特に下 5 つのルールは事前においてほとんどが不完全かなしであったが、事後ではありと不完全が多くなった。これらルールは繰り下がりにおいて数を借りる処理の手続きであり、事前ではこれらの説明がかなり欠落していたが、事後では説明記述が増えた。

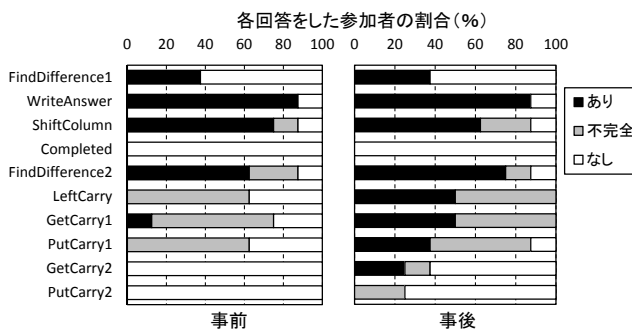


図 4: 手続き説明課題における各ルールの評価

バグ同定課題の平均点は事前で 0.88、事後で 1.13 であった。なお、システムを用いた実装モデルの具体化は、8 名全員が成功した。また、事前テストから事後テストを完了するまでにかかった時間は、おおむね 60 分程度であった。

4.3 考察

図 4 に示されるように、システムを用いて実装モデルへの具体化を経験することで、引き算の手続きの説明が

改善された。事前では、とりわけ繰り返し下がり処理の手続きが省略された説明が多かったが、モデル具体化後にはこれに関する説明がある程度なされるようになったということである。一方で図4の一番上のFindDifference1のように、事前・事後でほとんど変化が見られないものもあった。これは、「各桁の引き算を実行する時は引かれる数が引く数以上でなければならない」という条件である。さらに、事後でも多くの参加者の回答は不完全で、9008-3149のように2桁以上離れた桁から数を借りる必要がある計算を実行できる手続きにはなっていないかった。この点については、事後テストの報告で「自分が、思っているより、自分自身は、あの長いプロセスをこなしているんだなあと思った」や「物事の考え方を文面にして説明する時自分が勝手に省略してしまうことが多くあると思った」など、無意識化している手続きに関する気づきが5人の参加者の回答に見られた。このことは、無意識化された思考過程を外在化することの難しさを示しており、モデルの具体化を通じて多少の改善があったものの不十分であったということであろう。バグ同定課題については、事前から事後にかけて点数が向上した参加者は2名であったが、本実験ではバグモデルの学習を行っていないため、この課題の改善にまでは至らなかったと考えられる。

参加者がシステムを操作した際のログには、ルールをチェックした時のエラーが8人で合計63件記録された。このうち20件は、DoCoProの仕様についての周知不足によるもの²であった。残りの43件のうちの19件は、(上位桁 L R)³と記述すべき条件を(上位桁 R L)とするなど、変数の位置関係の誤りが原因であった。試作システムの学習活動は、プロダクションモデルのうちルールの具体化を含むものの、問題表現の設計は対象とせず、その解説を学習者に与えている。事後テストの報告では、「作成というよりも、作業になってしまいプログラミング自体の把握ができなかった」や「図を見てどのように推論されているのか見てみたい」といったように、問題表現の把握が困難であったと考えられる回答も含まれていた。この対策のひとつは、各ステップの状態を可視化することであるが、抽象モデルの具体化に焦点を置いた学習活動においては、問題表現を解釈する負荷を完全に取り除くべきでないかもしれない。そのため、問題表現を理解するための学習活動を加える必要があるだろう。

以上の結果より、本研究で提案するフレームワークにおける学習により、学習が起こる可能性はあると考えられる。ただし、今回試作したシステムは低い負荷でモデル構築を経験することを意図しているが、問題表現理解のための学習活動を加えることも含め、負荷

² ルールの名前に全角かっこを使用するとエラーが起こるというものである

³ 「桁 L は桁 R の上位桁である(L , R は変数)」という事実を表す述語である

を調整することが必要であろう。

謝辞

本研究は、科学研究費補助金15K12426の助成を受けた。

参考文献

- [1] Basu, S., Dukeman, A., Kinnebrew, J., Biswas, G., Sengupta, P.: Investigating Student Generated Computational Models of Science, *Proceedings of 11th International Conference of the Learning Sciences*, pp. 1097-1101 (2014)
- [2] Hirashima, H., Imai, I., Horiguchi, T., Toumoto, T.: Error-based simulation to promote awareness of errors in elementary mechanics and its evaluation, *Proceedings of 14th International Conference on Artificial Intelligence in Education*, pp. 409-416 (2014)
- [3] 神崎奈奈, 三輪和久, 寺井仁, 小島一晃, 中池竜一, 森田純哉, 齋藤ひとみ: 認知モデル作成による認知情報処理の理解を促す大学授業の実践と評価, *人工知能学会論文誌*, Vol. 30, pp. 536-546 (2015)
- [4] Kojima K., Miwa, K., Matsui, T.: Supporting Mathematical Problem Posing with a System for Learning Generation Processes through Examples, *International Journal of Artificial Intelligence in Education*, Vol. 22, pp. 161-190 (2013)
- [5] Miwa, K., Morita, J., Terai, H., Kanzaki, N., Kojima, K., Nakaike, R., Saito, H.: Use of a Cognitive Simulator to Enhance Students' Mental Simulation Activities, *Proceedings of 12th Intelligent Tutoring Systems*, pp. 398-403 (2014)
- [6] 中池竜一, 三輪和久, 森田純哉, 寺井仁: 認知科学の入門的授業に供するWeb-basedプロダクションシステムの開発, *人工知能学会論文誌*, Vol. 26, pp. 536-546 (2011)
- [7] 中島秀之: 構成的研究の方法論と学問体系, *Synthesiology*, Vol. 1, pp. 305-313 (2008)
- [8] Schunn, C. D., Crowley, K., Okada, T.: The Growth of Multidisciplinary in the Cognitive Science Society, *Cognitive Science*, Vol. 22, pp. 107-130 (1998)