

生成AIを利用したプログラミング学習手法の開発と実践

Development and practice of programming learning methods using generative AI

飯塚 康至[†]
Yasushi Iizuka[†]

[†]ビジネス・ブレークスルー大学 経営学部

[†]Faculty of Business Administration, Business Breakthrough University.

要旨

ChatGPT に代表される生成 AI は、テキストの生成だけでなく、プログラムの生成にも優れた能力を持っている。プログラムの生成は通常のテキスト生成とは異なり、生成されたプログラムを実行することで内容を検証できるため、プログラミング学習やプログラミング手法の向上に役立つと期待されている。本発表では、プログラミング初心者を対象に生成 AI を活用してプログラミングを行い、その過程を通じてプログラミングを学ぶ手法を開発し、実践した結果について報告する。

1. はじめに

ビジネス・ブレークスルー大学では 2022 年より社会人向けにコホートベース学習（飯塚, 2024）を取り入れた短期集中コースである履修証明制度（インテンシブコース）を開催している。インテンシブコースは全体で 3 ヶ月のコースである。コース内には 3 週間のプログラミングのセクションがある。このプログラミングのセクションは週 1 回のライブ講義（全 3 回）と成果物の提出、他者とのコミュニケーションから成り立っている。

プログラミングの取得時間は例えば Java などは 200 時間以上の学習が必要とされると言われていて、3 週間という短い学習期間では一からプログラミングの基礎を学んでいくことは不可能である。そこでプログラミングのセクションでは「パラシュート学習法」を取り入れている。パラシュート学習法とは山の頂上などの目標地点に空中からパラシュートで降りるような学習法であり、基礎から山登りをして学習し目的にたどり着くような従来の学習法とは異なり、まずは成果を手にするような学習法である。パラシュート学習法によってプログラムの文法に関する知識は主体とせず、プログラムを実行したり、プログラミングスキルを身につけるための知識やスキル（メタスキル）を身につけたりすることを目的としている。

従来のパラシュート学習法ではサンプルのプログラムを事前に用意し、サンプルのプログラムをコピー＆ペーストし、プログラムの実行を行うという流れで演習を行ってきた。そしてプログラムを改造しながら理解を深めていくという流れである。

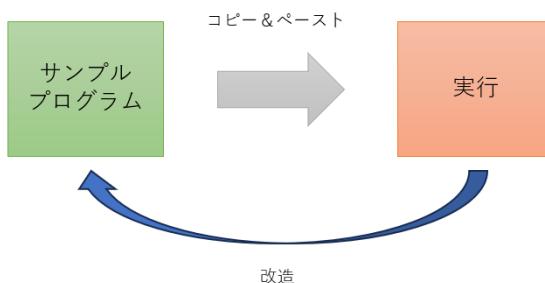


図 1 従来のパラシュート学習法

生成 AI の登場により、サンプルプログラムを用意することなく、学習者の指示でプログラムを生成し、生成したプログラムを実行することができるようになってきた。またプログラムの改造や解説を生成 AI に行わせることにより、自分のレベルに合わせた形で生成されたプログラムの理解を深めることができた。

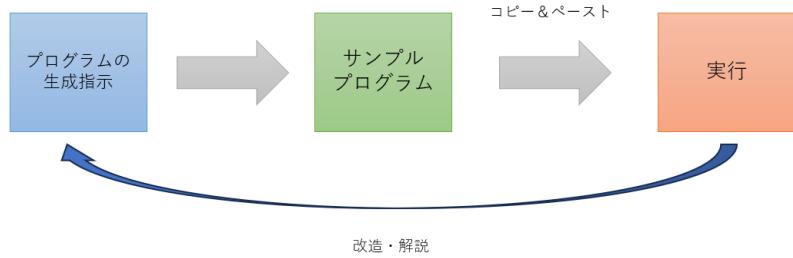


図 2 生成 AI を用いたパラシュート学習法

次節より実際に実践した生成 AI を用いたパラシュート学習法の詳細について述べる。

2. プラットフォームの選定

2.1. 生成 AI の選定

生成 AI は OpenAI 社が提供している ChatGPT のほかに ChatGPT をベースに Microsoft 社が提供している Copilot や Google 社が提供している Gemini, Anthropic 社が提供している Claude, Perplexity AI 社による Perplexity などがある。これらの生成 AI は各社の個性が反映されていて、出力結果が異なる場合があるし、ハルシネーションと呼ばれる間違った情報を出力する場合もある。特に固有名詞などではこの傾向が顕著である。

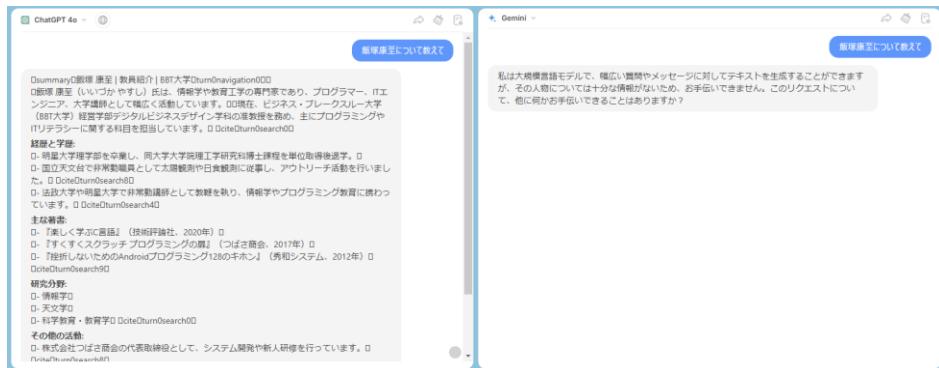


図 3 人物について問うたときの結果

ところがどの生成 AI も基礎的なプログラムの生成を依頼すると同じような結果を出力する。



図 4 Java のプログラムを出力した結果

生成 AI はほとんどの場合に無料で使えるようになっているが、学習者によって使っている生成 AI の種類が異なったり、有料のサブスクリプションを登録していたりと違いがあるので、生成 AI の選定については特に制限は設けていない。生成 AI を使ったことがないという学習者には ChatGPT の利用を進めている。これは ChatGPT が生成 AI を牽引してきたことと、ログインや登録なしに利用できるようになっているためである。

2.2. プログラム言語と実行環境の選定

プログラム言語と実行環境の選定は重要である。本稿の実践を行ったインテンシブコースはオンラインのライブ講義に学習者が集う形式であるため、学習者のPC環境がさまざまである。プログラムの実行環境をPCにインストールし初期設定を行う必要があるような言語（JavaやC言語）は利用に適さない。必然的にブラウザ上で実行可能なプログラム言語と実行環境を選定することになる。

インテンシブコースでは2つの言語、2つの実行環境を利用することとした。1つ目の言語は機械学習やデータサイエンスなどで人気のあるPythonであり、Pythonの実行環境としてオンライン上で実行可能なGoogle Colaboratoryを選定した。2つ目の言語はJavaScriptをベースに発展したGAS(Google Apps Script)であり、GoogleスプレッドシートなどのGASの実行環境である。

2つの言語を2つの実行環境で同時に学ぶのはPythonとGASの言語体系の違いを確認する意味もあるが、言語により得意不得意があることを学習する意味合いもある。

生成AIにより指示を出せばプログラムが生成されるが、言語の指定により簡単に生成できる場合と簡単に生成できない場合があることを知るためである。

3. 実施例

プログラミングの生成における指示文（プロンプト）は基本的に次のような形になる。

「【実行環境または言語名】で実行可能な【やりたいこと】のプログラムを生成して」

簡単な例を上げて実際の結果と学習者が得られるメタスキルやメタ知識を紹介する。

3.1. Google Colaboratory の場合

プロンプト：「Google Colabで実行可能なハート図形を描くプログラムを生成して」

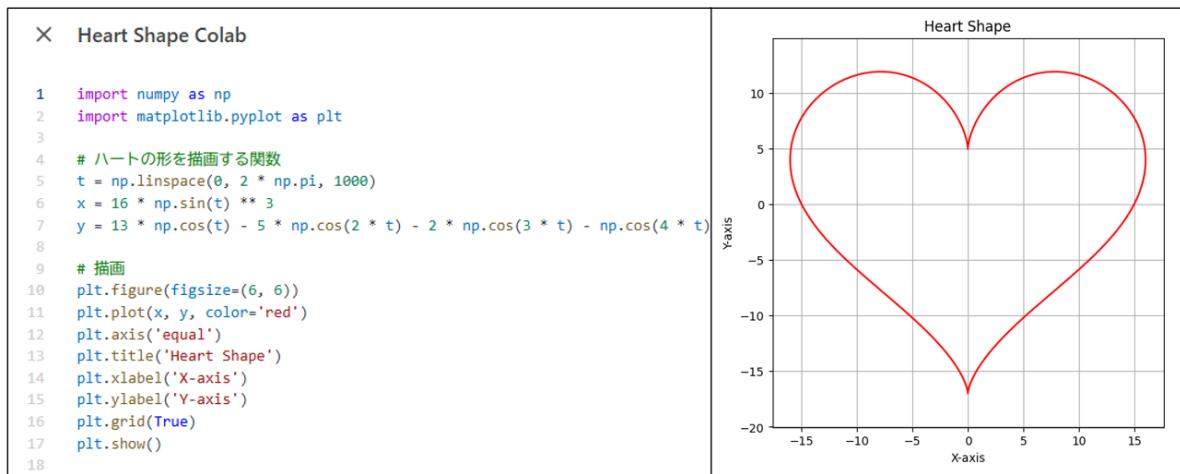


図5 出力されたプログラムと実行結果（Google Colab）

プログラムの生成から実行までの流れを通じて学習者は次のようなことを学び、独自のプログラムの生成（改造）を行っていく。

1. プログラムの出力からGoogle Colabでの実行までの方法
2. 同じ指示でも完全に同じ出力にはならないこと
3. プログラムはあっていいるがGoogle Colabで実行できないプログラムが生成される場合があること
4. ハートの図形にならない場合があること
5. 日本語がタイトルなどに含まれていた場合は文字化けし、この解消は比較的大変なこと
6. プログラムの知識以外に数学の知識などが必要なこと

学習者これらのメタスキルやメタ知識を学び、ハートの図形外に改造しながらプログラムに慣れていくことになる。

3.2. Google Apps Script の場合

プロンプト：「GASで実行可能な自分にメールを送信するプログラムを生成して」

The screenshot shows a split-screen view. On the left, a code editor displays a Google Apps Script (GAS) code:

```

1 function sendEmailToSelf() {
2     var recipient = Session.getActiveUser().getEmail();
3     var subject = "自分へのメールテスト";
4     var body = "これはGASから送信された自分自身へのメールです。";
5
6     MailApp.sendEmail(recipient, subject, body);
7 }
8

```

On the right, the execution result is shown in a simulated email interface:

自分へのメールテスト > 受信トレイ ×

To: 自分

日本語 日本語 メールを翻訳

これはGASから送信された自分自身へのメールです。

返信 転送

図 6 出力されたプログラムと実行結果 (GAS)

GAS のプログラムでも Google Colab のときと同様に学習者は実行までの流れと、次のようなことを学び、プログラムの改造を行っていく。

1. セキュリティ関係の権限の付与を適切に行わないと実行できること
2. 送信できない場合、プログラムの知識以外にネットワークの知識が必要なこと
3. JavaScript をベースにした GAS の場合、記号や構造に意味があること

4. プログラム学習のための指示例（プロンプト例）

パラシュート学習法を利用してプログラムの生成から実行までの流れを通じてプログラミング学習におけるメタスキルやメタ知識を得る他に生成 AI を利用してプログラムの理解を深めることもできる。例えば、生成 AI はプログラムの生成とともに実行までの手順などを解説してくれるが、プログラムにコメントを埋め込むように指示することでプログラムの各命令が何を行っているか分かりやすくすることができる。

また、「小学生にわかるように」や「料理のメタファー（比喩）」でプログラムを解説してもらうなどの工夫により学習者のそれぞれのレベルに最適化した形でプログラムの解説を生成することもできる。

5. 考察とまとめ

生成 AI を用いたプログラミング学習法の開発と実践を行ってきた。得られた知見をまとめ本稿の結びとしたい。

- 1) パラシュート学習法において生成 AI の活用は有用である
- 2) 生成 AI への指示文を工夫することで生成されたプログラムの理解を深めることができる
- 3) プログラミング学習ではプログラム自体と同じくらい実行環境などの周辺知識への理解が重要で生成 AI を利用したパラシュート学習法ではこれらメタスキルやメタ知識を得ることができる

参考文献

- [1] 飯塚 康至 (2024), コホートベース学習コースの設計と運用, 日本教育工学会 2024 年秋季全国大会 (第 45 回), 2024/9/7
- [2] 原田 紗希 ほか (2024), 生成 AI を用いたペアプログラミングによるプログラミング自己学習方法の開発, 日本教育工学会論文誌, 論文 ID: S48115
- [3] 村田 美友紀 ほか (2024), 生成 AI によるプログラミング教育のパラダイム転換と教育支援ツールに関する研究構想, 情報教育シンポジウム論文集, 2024, 95 - 102