

非英語圏の国における IT 研修の新たなアプローチ： 母語音声生成と英語表記を活用したプログラミング学習の効果

白川清美¹ 千葉亮太¹ 石田三智²

A Novel Approach to IT Training in Non-English-Speaking Countries : The Effectiveness of Programming Education Using Native Language Speech Generation and English Notation

SHIRAKAWA Kiyomi¹, CHIBA Ryota¹, ISHIDA Michi²

本研究では、非英語圏の国（カンボジア及びベトナム）を対象とした IT 研修において、新しい教育アプローチを提案する。具体的には、教授者が学習者の母語（クメール語やベトナム語）を理解していない場合でも効果的な指導が可能となるように、Python と各種ライブラリを活用して母語音声を生じ、英語表記と組み合わせた学習教材を作成した。

母語音声は学習者の直感的な理解を促し、英語表記は教授者と学習者間の共通理解を支援する役割を果たす。研修の対象は、SAS, Excel VBA, Python を学ぶ初級者とし、音声付き動画とあわせて各自でプログラムを動作させる実習形式を採用した。なお、音声付き動画の作成では、画像やテキストから音声を生じする Python スクリプトを活用し、省力化・自動化を実現する仕組みを構築した。その結果、学習者のレベルに応じたリッチコンテンツ型の学習教材を、従来よりも柔軟に作成できるようになった。

本稿では、カンボジアにおける研修事例をもとに、母語音声生成と英語表記を組み合わせた教材の効果を分析し、学習成果の向上を確認した。さらに、今後は非英語圏であるベトナムにおいても同様の IT 研修を実施し、カンボジア以外の国にも適用可能であることを実証する予定で進めている。

Abstract

This study proposes a novel educational approach for IT training in non-English-speaking countries, with a focus on Cambodia and Vietnam. The approach leverages Python and various libraries to generate native language speech while incorporating English notation to develop instructional materials. This method enables effective teaching even when instructors do not understand the learners' native language (Khmer or Vietnamese).

By integrating native language speech, learners can develop a more intuitive understanding, while English notation serves as a common reference point between instructors and learners. The training program targets beginners studying SAS, Excel VBA, and Python, employing a hands-on approach in which participants execute programs while following voice-guided instructional videos. To enhance efficiency and automation in content creation, a Python script was implemented to generate speech from images and text, allowing for the streamlined

¹: 立正大学, ²: ホーチミン市技術師範大学

production of voice-supported learning materials. Consequently, this approach facilitates the development of rich, adaptable learning resources tailored to learners' proficiency levels.

The effectiveness of this method was analyzed based on a case study conducted in Cambodia, demonstrating improved learning outcomes. Future research aims to extend this IT training model to Vietnam to further validate its applicability across other non-English-speaking countries.

キーワード：母語対応教育, マルチリンガル教材, 音声合成技術, プログラミング教育デザイン, インクルーシブ学習

1. はじめに

カンボジアにおける IT 教育の背景

カンボジアにおける IT 教育の普及は、多くの課題がある。その要因の一つとして、カンボジアの公用語であるクメール語が Windows OS の正式な対応言語として提供されていないことが挙げられる。現在、Windows OS が対応しているアジア言語は、日本語、中国語（簡体字・繁体字）、韓国語、タイ語、ベトナム語に限られており、カンボジアでは英語版の OS を使用せざるを得ない状況にある。これにより、英語の習熟度が低い学習者にとって、IT 関連の知識を習得することが大きな障壁となっている。

本研修のメインターゲットであるカンボジア国家統計局（NIS: National Institute of Statistics）においても、IT 教育の困難さが顕著に表れている。カンボジアでは統計調査の実施件数が少なく、統計業務に従事する職員は高度なプログラミングスキルを日常的に求められることは少ない。そのため、統計データの集計や分析に関する専門的な知識を有する職員は限られており、主な業務はデータ入力にとどまっている。このような状況では、プログラミング教育の必要性が十分に認識されにくく、学習環境の整備も進みにくい。

さらに、カンボジアにおける IT 教育の発展が遅れている要因の一つに、国際支援プロジェクトの課題がある。著者は、2005 年に開始された JICA（国際協力機構）の統計支援プロジェクトに参加し、カンボジアにおける IT 研修を実施してきた。さらに、スウェーデン国際開発庁（SIDA）や国連人口基金（UNFPA）も統計分野での支援を行っているが、IT スキルの顕著な向上は見られない。その一因として、支援プロジェクトごとに使用するプログラミング言語が異なることが挙げられる。これにより、統一された学習環境が確立されず、学習者がプログラミングスキルを体系的に向上させることが困難となっている。

しかし、近年の生成 AI 技術の発展により、さまざまなメディアを活用した研修教材の作成が容易になってきている。従来は教材の開発に多くの時間とリソースを要していたが、生成 AI を活用することで、学習者のスキルレベルに応じた教材を迅速に作成できるようになった。これにより、個々の学習者に最適化された IT 教育の提供が可能となり、従来の教育課題の一部を解決する手段として期待されている。

目的

本研究においては非英語圏の国における IT 教育の効果を高めるための新たなアプローチを提案する。NIS において、英語だけでなくカンボジアの母語（クメール語）でも教材を作成し、多様なメディアを活用することで、学習効果の向上を目指す。非英語圏の国における IT 研修では、教授者と学習者の共通言語が英語であるため、両者には英語でのコミュニケーション能力が求められる。しかし、英語の習熟度が低い学習者にとって IT 教育を効率的に行うことは難しい。そこで、既存の英語教育の手法や母語のメディアを活用し、IT スキルの向上としてプログラミング言語の学習効果をさらに高めることを目指す。

また、生成 AI を活用することで、学習者のスキルレベルに応じた教材を迅速に作成・提供し、個々の理解度に合わせた学習を支援する。このアプローチにより、従来の統一的なカリキュラムでは対応が難しかった学習者ごとのニーズに適応し、柔軟な研修を実現する。結果として、より効果的な IT 教育の提供が可能となる。

これらをカンボジアにおける IT 研修において、学習者が高度な統計処理やデータ分析を効率的に学習できる教材を整備し、本研修の目的が達成できるようにする。本論文における研修のプログラム言語として、SAS、Python、Excel VBA の 3 つの言語を選定した。これは、SAS の強力なデータ処理機能により CPI（消費者物価指数）などの精度が求められる計算で活用が可能なおことから、公的統計集計や分析に用いる中心的な言語とする。加えて、Python は SAS では対応できない高度なデータ可視化や多くのライブラリがあることからデータの前後処理、SAS や Excel との連携などに使用する。最後に、Excel VBA は、ガジェットや報告書作成における集計や可視化を効率化するために用いる。このように各言語の特徴を活かすことで、カンボジアの統計業務に適した実践的なスキルの習得を促進する。

2. 先行研究等

先行研究の参照と本研究との関連性

本研究では、母語が非英語圏であるカンボジアの NIS の職員を対象とした IT 研修において、学習者の母語音声生成と英語表記を活用したプログラミングの学習効果を検証する。プログラミングの学習においては、語学学習と共通する要素を持つと考えた。特に、視覚情報、音声情報、動画を活用した学習方法の効果に関する先行研究を参照することで、本研究の手法の妥当性を示すことが可能となる。

① 視覚情報の役割とプログラミング学習

九州大学の李相穆（2019）の研究「外国語教育における視覚情報の効果」は、視覚情報（静止画、動画）が語彙学習の効果を向上させることを示した。特に、イメージ・スキーマの形成が語彙理解を助けることが示唆されており、本研究で提案する「母語音声と英語表記の組み合

わせ」も、学習者のプログラムコードの理解を促進する可能性がある。さらに、動画による動作説明が記憶定着を強化するという点も、自身がプログラム実行時にリアルタイムで動作の説明を受けながら実行することと関連する。

② 音声の活用と英語プログラミング学習

中池宏行（2018）の「学習方法の違いによる語彙習得率の比較研究」では、音声を用いた語彙学習がリスト暗記と比較して効果的であるかを調査した。その結果、音声の付加自体は単語の習得率に大きな影響を与えなかったが、視覚情報と組み合わせることで理解が深まることが示唆されている。

本研究の「母語音声を用いたプログラミング学習」においても、音声単体ではなく英語表記と併用することで、学習者がコードの概念をより直感的に理解できると考えられる。特に、「英語表記と併用」することで、機械翻訳における誤訳の影響を軽減できる。例えば、カンボジアにおける事例では、「Windows」という単語が Microsoft 社の PC の OS を指すにもかかわらず、クメール語では「窓」と誤訳されることがあった。このような誰もが認識している単語に限らず、専門的な技術用語においても、正確な理解を促すために英語のオリジナル表記を併用することが求められる。

③ 英語ライティングとプログラミング学習

成田真澄（2018）の「日本人英語学習者の英語ライティングを伸ばす指導の方向性」では、日本人学習者のライティング指導の課題を指摘し、論理的な構造化が重要であることを述べている。プログラミング学習も同様に、定められた構文に基づいて適切な表記法が求められること、正しい結果を導くためには構造的な理解が必要であることから、新たな言語を学ぶ（非英語圏における英語を習得する）場合と同様に考えることが可能となる。加えて、英語表記と併用することが「コードの理解の向上」と「構文の理解促進」に寄与する可能性がある。

プログラム言語の習得との関係

本研究では、非英語圏の国を対象とした IT 教育において、学習者の母語音声と英語表記を組み合わせた新たな学習手法を導入し、その効果を分析する。プログラム言語の習得には、研修テキスト等に記述されたコードの構文や意味を正しく理解する能力が求められるが、英語を母語としない学習者にとって、英語のみの教材では十分な理解を得ることが困難である。そこで、生成 AI を活用することで、学習者のスキルレベルに応じた追加教材を迅速に作成し、個々の理解度に合わせたテキストを元に、母語音声を生成し、英語表記と組み合わせた教材を活用することで、教授者と学習者の円滑なコミュニケーションと学習効果の最大化につなげ学習効果を高めることを本研究の基盤とする。

① 学習者の母語音声と英語表記を活用する手法の優位性

プログラミング言語は基本的に英語を基盤として設計されており、例えば、予約語や構文、ライブラリの変数名及び関数名などは英単語で構成されていることが多い。そのため、非英語圏の学習者にとっては、プログラムの概念を理解することに加え、言語としての英語も理解する必要があり二重の負担となる。また、英語におけるリーディングやライティングの際にも、会話や物語には論理的な順序が存在し、これはプログラムにおけるアルゴリズムの構造に相当する。したがって、プログラムを理解するには言語における文法的な学習とアルゴリズムなどの構造的な学習の両方において、それぞれの概念を体系的に理解することが求められる。

本研修では、学習者の母語音声と英語表記を組み合わせたスライドや教科書を使用し、これらの概念の習得を支援することで、理解の負担を軽減し、学習の効率を向上させることを目指す。さらに、教授者が学習者の母語を理解しない場合であっても、英語表記を共通言語とすることで円滑な指導が可能となる。従来の研修では、教授者が英語のみで説明し、学習者がその英語を理解できなかった場合、学習進度に遅れが生じる課題があった。しかし、母語の音声解説を同時に提供することで、学習者は自身の言語で直感的に内容を理解できる。この手法により、学習のスムーズな進行が可能となり、英語理解のレベルによらず、より多くの学習者がプログラム言語を習得できる環境が実現する。

② 生成 AI による母語音声教材が学習者の理解に与える影響

近年、生成 AI の発展により、テキストからリアルタイムで母語音声を生成することが可能となった。これにより、研修教材や教科書に英語表記とともに母語音声を付加することが容易となり、学習者の理解を支援する新たな手法として活用できる。特に、プログラムコードや概念の説明を母語で提供することで、学習者がコードの構造や実行結果をより深く理解できる。

さらに、関連研究にあるとおり、視覚的な教材（スライドや教科書）と音声情報を併用することで、学習者は複数の感覚を用いて知識を習得できるため、記憶の定着率が向上すると考えられる。このアプローチは、特に英語表記に対する抵抗感がある初学者に対して有効であり、プログラム言語の学習をスムーズに進めるための重要な要素となる。

③ 英語表記を通じた教授者と学習者間の共通理解の形成

プログラム言語の習得において、教授者と学習者間の共通理解を形成することは極めて重要である。本研修では、英語表記を研修の標準言語とすることで、母語が異なる学習者と教授者が共通の理解を持てる環境を構築する。英語表記があることで、教授者がプログラムコードの説明をする際に、学習者が理解しやすくなるだけでなく、学習者が自身でコードを調査し、学習を進める際の指標となる。

3. 研修設計と教材作成手法

本研究では、「複数のメディアに対応するための Python スクリプト」を活用し、講義資料やスライドのテキストを手間がかからず短時間で母語音声・動画生成する仕組みを構築する。この仕組みを採用した本研修では、NIS の学習者に対し、従来の教材開発では対応が難しかった個別指導のための教材作成を行い、より柔軟で効果的な学習環境を提供する。先述したとおり、英語を教授者と学習者の共通理解の基盤とし、教授者が学習者の母語を理解できなくても、生成された教材を通じて効果的な指導が可能となる。学習者は英語字幕を通じてコードの意味を把握しながら、母語音声による解説を同時に聞くことが可能となる。

この手法は、英語を理解できる学習者にとっては、母語よりも英語を用いることで学習効果が高いことを考慮するとともに、特に、英語に不慣れな初学者にとってはプログラムコードの解説を学習者の母語で提供することで、学習の負担を軽減し、学習の障壁を下げる効果が期待できる。

母語音声・動画生成の技術的詳細

本研究では学習者の母語音声及び英語字幕を付与した動画を生成する手法を提案する。具体的には、以下に掲げる一連の処理を Python スクリプトにより自動化することで、講義資料やスライドのテキストを省力的に生成する仕組みを構築する。教材の種類に応じて、以下の複数の生成 AI ツールを組み合わせる（表 1 参照）。

なお、複数のメディアに対応するための Python スクリプトについても生成 AI を活用し作成した。

① テキストの生成

既存の PPTX ファイルや画像を利用し、それぞれの画像（PPTX ファイルの場合はスライド）に適した説明文をテキストとして作成する。なお、教授者は学習者のスキルレベルを適切

表 1 メディア別に使用する生成 AI

メディア別教材	使用する生成 AI	役割
テキスト教材 (スライド・教科書)	ChatGPT, NotebookLM	学習者のレベルに応じたテキスト生成・整理
動画教材（英語字幕＋母語音声）	Gemini	AI を活用したスクリプト生成、文章要約、字幕作成支援
	Canva	AI を活用した画像生成、テンプレートを活用したスライド・動画編集、画像を組み合わせた動画作成
音声教材（母語音声ガイド）	Google TTS, Azure Speech	母語音声の自動生成、ナレーション作成

に把握し、それに応じた教材を用意することが重要である。教材には文字情報だけでなく、画像やインフォグラフィックを取り入れることで、視覚的な補助を強化し、理解を深める構成とする。これらを入力として、ChatGPT を活用して英語や母語のプログラム解説を作成し、NotebookLM などのコンテキスト（文脈・背景）情報を与えることのできるサービス・ツールを活用し、より文脈に適した解説のテキストなどの自動生成を行う。

② 母語音声・動画の作成

生成した解説のテキストを Google Cloud Text-to-Speech, Azure Speech API などの音声合成技術を用いて、学習者の母語に対応した音声を作成する。Canva を用いて視覚的に分かりやすいスライドを作成し、それを動画に組み込む。Gemini を活用し、スキルレベルに応じた適切な動画シナリオを作成し、Canva の動画編集機能と統合する。

③ 字幕と音声の統合

英語字幕を Whisper で自動生成し Canva などの GUI による操作や Python の ffmpeg のようなライブラリを活用して、音声と同期した字幕付き動画を作成する。これにより、英語字幕と母語音声統合された学習教材を提供できる。

以上のように、生成 AI を駆使し、それを一連の流れを Python スクリプトで記載することで、従来難しかった教材の個別最適化が容易になり、柔軟な IT 教育が実現可能となる。また、この流れを図示化すると、図 1 のとおりとなる。

さらに、図 1 のプロセスで作成される PPTX ファイル（スライドとクメール語）は、図 2 に示すとおりである。

今後の展開としては、生成 AI の精度向上や学習データのカスタマイズにより、より効果的な母語音声教材の開発を進めるとともに、学習者のフィードバックを反映した教材改良を行うことが求められる。

4. 研修事例と結果分析

カンボジアにおけるこれまでの研修と研修実施事例

カンボジア国家統計局（NIS: National Institute of Statistics）において、2005 年から 2013 年まで、Microsoft SQL Server, VB. NET, ACCESS, Excel VBA, R などの研修が実施され、これらの研修は、NIS が実施する統計調査（主に国勢調査や経済センサス）の実施に合わせ、実務での活用を前提としており、データ管理及び統計分析のスキル向上を目的とした。特に Excel VBA や R は、統計集計やデータ処理の自動化に活用した。

2015 年以降 2023 年までは、一橋大学経済研究所及び立正大学の研究支援のもと、従来と同様の研修を NIS 職員に対して行われた。この期間においても、Excel VBA や R を中心に、

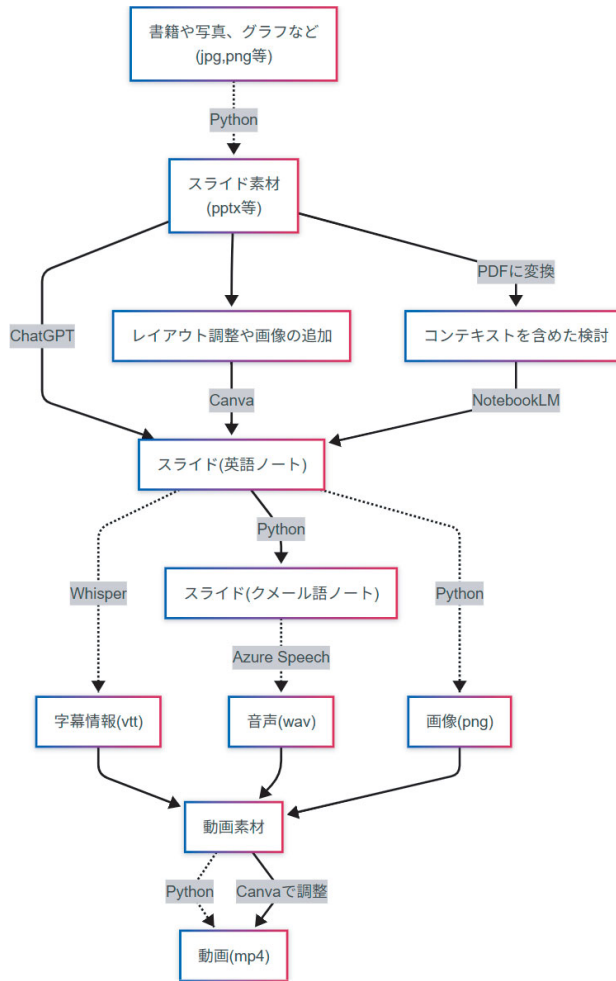


図1 動画作成のフロー

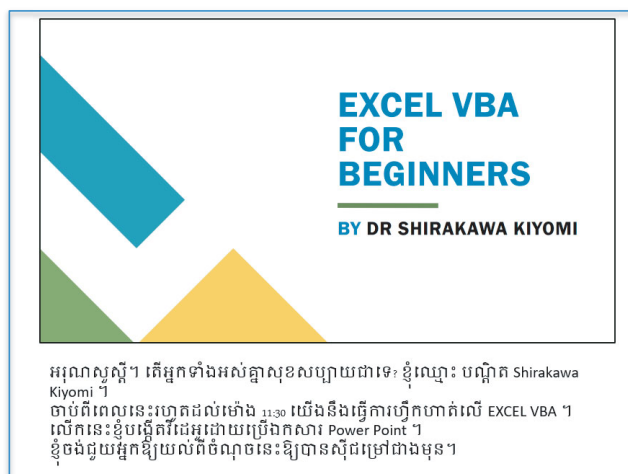


図2 動画作成のための PPTX ファイル（スライドとクメール語）

NIS 職員のデータ分析及び統計処理能力の向上を目指した教育が実施された。

2024 年には、Excel VBA と Python を中心とした研修を実施したが、12 月の研修においては、Excel VBA に加え、新たに SAS の研修を実施した。SAS は、統計解析及びデータ管理に優れた機能を持つため、今後の NIS の統計業務において重要な役割を果たすことが期待される。

なお、NIS の統計調査の詳細については、NIS 公式サイトを参照されたい。

研修成果への期待

現在、NIS に対する SIDA（スウェーデン国際開発庁）や JICA（国際協力機構）の支援は終了しており、NIS 独自の体制で IT 研修を推進する必要がある。そのため、NIS 内での統一されたプログラム言語のスキル向上が課題となっている。

著者は、2024 年 8 月、National School of Planning and Statistics (NSPS) において、Excel VBA の講義を実施している。

さらに、2025 年以降は、SAS を NIS での統一プログラム言語とし、そのスキル向上を目指す研修を NIS や NSPS での実施を計画している。SAS は、統計処理やデータ分析に適した高度な機能を持ち、NIS の統計業務における簡素化や省力化と精度の向上に寄与すると考えられる。

この SAS 研修では、本研究の技術を用いることで、学習者のスキルに応じて、複数のメディア（テキスト、音声、動画）を活用した研修資料を提供することにより、学習者が自身のレベルに合わせて学習を進めることが可能となり、研修の効果を最大化することが期待される。

研修の成果

2024 年 8 月に実施された Excel VBA 研修では、英語の理解が十分でない学習者に対し、クメール語の通訳による補助説明を加えることで、研修の進行を支援していた。しかし、同年 12 月に実施された Excel VBA 研修では、本研究の成果である複数のメディアを活用した母語教材を提供したことにより、クメール語の通訳が不要となった。

この結果、12 月の研修では、英語から母語への通訳にかかる時間が不要となり、同一の研修日程ではあったが、従来の研修時間の約 2 倍の時間を確保できるようになった。そのため、研修の効率が大幅に向上した。さらに、動画教材の導入により、学習者は予習及び復習が可能となり、理解度の向上が見られた。従来は、講義の進行に合わせて説明が行われていたが、動画を活用することで、学習者は必要に応じて内容を繰り返し学習できるようになった。

複数のメディアによる学習者の満足度と成績評価結果

2024 年 8 月と 12 月に実施した Excel VBA の研修において、学習者の満足度調査及び成績評価を行い、以下の結果を得た。特に 12 月の研修においては、母語音声と英語字幕を組み合わせた動画教材を活用することで、学習者の理解度の向上が顕著に見られた。

① 学習者の満足度

研修後のアンケート調査では、学習者の 85% が「研修の進行がスムーズになった」と回答した。さらに、80% の学習者が「動画教材の導入が理解度の向上に役立った」と評価している。従来の講義形式と比較し、事前学習が可能となったことで、講義中の理解が深まったことが主な要因と考えられる。

また、英語字幕と母語音声を組み合わせた教材については、90% の学習者が「より理解しやすくなった」と回答しており、特に英語の読解に不安を感じる学習者にとって、母語音声の追加が有効であったことが示された。

② 成績評価の結果

研修の成果を数値的に評価するため、2024 年 8 月と 12 月の研修前後でテストを実施し、成績の比較を行った。その結果、以下の改善が見られた。

8 月研修のテスト平均点：60 点（100 点満点）

12 月研修のテスト平均点：78 点（100 点満点）

特に、英語の理解が十分でない学習者の成績向上が顕著であり、平均点の向上率は 30% に達した。この結果は、複数のメディアを活用した研修資料が学習者のスキル向上に寄与したことを示している。

5. まとめと今後の展望

本研究では、カンボジア NIS における IT 研修の成果を分析し、複数のメディアを活用した研修資料の導入が、学習者の理解度及び成績向上に寄与することを示した。特に、英語字幕と母語音声を組み合わせた教材の提供が、学習の効率を向上させることが確認された。

今後の研修においては、SAS を NIS の統一プログラム言語とし、さらに高度なデータ分析スキルの習得を目指す。さらに、生成 AI を活用し、学習者のスキルレベルに応じた個別対応型の教材を迅速に作成することで、より効果的なプログラム言語の習得を支援する。特に、動画教材の拡充や、リアルタイムで学習進捗を把握できるシステムの導入を検討し、さらなる学習効果の向上を図る予定である。加えて、収集された研修結果のデータ分析を行い、プログラム言語の習得において必要な素養を持つ人材の特定を進めるとともに、学習者ごとに最適化された教材の提供方法をさらに発展させる必要がある。特に、SAS を中心とした統計プログラ

ムの教育において、多言語対応教材がどのような影響を与えるのかを引き続き検証し、プログラミング教育への応用を進めることが求められる。さらに、生成 AI を活用し、学習者のスキルレベルに応じた柔軟な教材提供の可能性を探ることで、より効果的な IT 教育モデルの確立を目指す。

多言語対応教材の有効性と課題の検証

本研究では、多言語対応教材を活用することで、教授者が学習者の母語を理解できない場合においても、母語が異なる学習者と教授者が共通の理解やコミュニケーションにおける障壁をどの程度克服できるのかを検証した。英語字幕と母語音声を組み合わせた動画教材や、生成 AI を用いた多言語対応のテキスト教材が、学習者のプログラム言語の理解に与える影響をより詳細に分析することにより、非英語圏の学習者に適した学習環境の構築を行った。加えて、多言語対応教材の導入による学習効果の向上が、どのような条件下で最大化されるのかを明らかにし、教材開発の改善点を特定することも本研究の重要な目的の一つである。

しかし、多言語対応教材には課題も存在する。教材の翻訳の精度や、音声合成の品質、さらには教材の作成にかかる負担が、学習の効果にどのように影響を与えるのかを検討する必要がある。特に、学習者の満足度と学習効果の関係を明確にすることで、多言語対応教材がどのように設計されるべきかの指針を得ることが可能となる。本研究では、これらの課題を整理し、今後の教材開発に向けた改善策を提案する。

また、今回の研修の実施によって、英語と母語の研修教材を提供したところ、英語の理解が十分でない学習者の成績向上が顕著であったことから、英語を母語としない学習者にとっては、英語の読解スキルが大きな障壁となり得ることなどの検証が期待できる。これは、プログラムコードの記述構造が英語の文法構造（主語・述語・目的語の関係や論理的な接続詞の使用）と類似している点が、英語スキルとプログラム言語の習得能力の関連性を示唆している可能性があると考えられる。

そのため、多言語対応教材を用いた学習がプログラム言語の習得に与える影響を検証するために、まず、英作文能力とプログラム言語の習得との相関関係を分析する。学習者の英作文能力（文法力、語彙力、論理的構合力）がプログラム言語の習得速度や正確性に与える影響を比較することで、英語スキルがどの程度プログラム学習に貢献しているのかを明確にする。次に、英語字幕と母語音声を用いた教材の効果を検証し、英語のみの教材と比較して、学習者の理解度が向上するかを分析する。さらに、母語でのプログラム解説がコードの構造理解にどのように寄与するかを評価し、生成 AI を活用した多言語対応テキスト教材が、学習者の復習や知識の定着率を向上させるかを調査する。

プログラミング教育への応用可能性

上記の考察の結果に基づき、どのような素養を持つ人材がプログラム言語の習得に適しているかを事前に特定することが可能となる。特に、英語の構文理解力が高い学習者は、プログラムコードの構造を素早く把握し、エラー修正の能力にも優れることが考えられる。さらに、論理的に正確な英作文を作成できる学習者は、コードの流れを理解しやすく、プログラムの実装においても論理的な整合性を維持しやすい。さらに、母語と英語の両方でプログラムの概念を学ぶ学習者は、異なる言語間の対応関係を理解する能力が高く、プログラム言語の習得においても柔軟性を発揮する可能性がある。

これらの分析をもとに、プログラミング教育への応用可能性を検討する。事前に学習者の素養を評価し、適切な教材や指導方法を提供することで、より効果的なプログラミング教育が実現可能となる。英作文能力が高い学習者には、英語ベースのプログラム言語の教材を重点的に提供し、より高度なスキルの習得を促す。しかし、英作文に不安のある学習者には、母語を活用した教材を提供し、基本概念の理解を支援する。さらに、生成 AI を活用することで、学習者ごとに最適化された教材を自動生成し、個別の学習ニーズに対応する可能となる。

また、プログラム言語のスキル向上に向けた教育方法の改良にも寄与する。特に、どのような素養を持つ人材がプログラム言語の習得に適しているかを事前に特定することで、より効果的にプログラム言語を習得できるのか、習得に時間がかかる受講生との違いを検討することで、より細やかな指導が可能となる。これを実現するために、学習者の英作文スキルを指標として、プログラム言語の習得スピードや理解度を予測することで、適切なカリキュラムの設計や指導法の選定が可能となる。

これらを調査するため、日本の大学生を対象に、英作文スキルとプログラム言語の習得能力の相関を検証する。さらに、今後はカンボジアやベトナムからの交換留学生（大学生）を対象として、英語スキルとプログラム言語の習得能力についても分析を行う予定である。これにより、異なる言語背景を持つ学習者におけるプログラム言語の学習過程を詳細に比較し、最適な指導方法を導き出すことを目指す。

今後の展望

本研究の成果は、共著者が所属するベトナムの大学の日本語学科の教育にも応用される予定である。カンボジアとベトナムの間で、プログラム言語の学習環境に違いを生じさせる要因として、母語の Windows OS の有無が影響を及ぼす可能性もある。カンボジアでは、クメール語の Windows OS が提供されていないため、英語版 OS の使用が標準となっている。一方、ベトナムでは、ベトナム語対応の Windows OS が存在するため、学習者は母語環境でコンピュータ操作が可能である。この違いが、プログラム言語の学習においてどのような影響を与えているのかについても検証を行う。

最後に、プログラム言語の学習において複数のメディアを活用することが学習成果の向上に寄与する場合、その手法を日本語教育にも適用し、言語習得プロセスにおける効果を検証する。多言語対応教材の導入が、プログラム言語の学習だけでなく、日本語の学習にも有効であるかを評価することで、技術教育と語学教育の双方において、より効果的な教育モデルを確立することが期待される。

他地域での応用を目指した改善提案

各言語の習得において、効果的なメディアの選択は学習者の効率に多大な影響を及ぼすため、以下に、日本語、英語、フランス語、スペイン語、中国語、韓国語、カンボジア語、ベトナム語の学習において優先すべきメディアをまとめた（表2参照）。

表2 言語別優先すべきメディアとその根拠

言語	優先すべきメディア	根拠
日本語	文字＋音声	日本語は漢字、ひらがな、カタカナの三種の文字体系を持ち、音声と組み合わせることで効果的な学習が可能である。
カンボジア語	文字＋音声＋画像	カンボジア語（クメール語）は独自の文字体系を持ち、音声と文字の組み合わせに加え、画像を用いることで理解が深まる。
ベトナム語	文字＋音声	ベトナム語はアルファベットを基盤とし、声調があるため、音声と文字の組み合わせで効果的な学習が可能である。
英語	文字＋音声	英語はアルファベットを基盤とし、音声との組み合わせで発音とスペルの関係を理解しやすい。
フランス語	文字＋音声	フランス語もアルファベットを使用し、特有の発音が多いため、音声と文字の併用が効果的である。
スペイン語	文字＋音声	スペイン語は発音とスペルが比較的一致しており、文字と音声の組み合わせで効率的に学習できる。
中国語	文字＋音声＋画像	中国語は漢字を使用し、声調が重要なため、音声と文字の組み合わせに加え、画像を用いることで意味理解が深まる。
韓国語	文字＋音声	韓国語はハングル文字を使用し、音声と文字の組み合わせで効果的な学習が可能である。

(参考 1)．複数のメディアに対応するための Python スクリプト（事例）

以下に示す図 3 から図 6 の Python スクリプトは、Python を活用して母語音声を生じ、英語表記と組み合わせたリッチコンテンツ型の学習教材を作成するためのものである。

なお、これらのスクリプトは生成 AI を活用したコーディングによって作成されたものを、特定の目的に応じて実行可能な形に調整を行っている。

```
from PIL import Image
import os

# フォルダパスと出力ファイル名の設定
folder_path = r" PDF の保存先フォルダ名 " # 画像が保存されているフォルダ
output_pdf = r" PDF の保存先フォルダ名と PDF ファイル名 "
# ファイル名の範囲を定義
start_page = 1
end_page = 500

# ファイル名を作成し、昇順でリスト化
png_files = [f"screenshot_page_{i}.png" for i in range(start_page, end_page + 1)]

# 有効な画像ファイルをチェックして読み込み
image_list = []
for file in png_files:
    img_path = os.path.join(folder_path, file)
    if os.path.exists(img_path): # ファイルが存在する場合のみ処理
        img = Image.open(img_path).convert("RGB")
        image_list.append(img)

# 最初の画像を基準に PDF を作成
if image_list:
    image_list[0].save(output_pdf, save_all=True, append_images=image_list[1:])
    print(f"PDF ファイルが作成されました: {output_pdf}")
else:
    print("有効な画像ファイルが見つかりませんでした。")
```

図 3 画像ファイルの PDF 化


```
from pptx import Presentation
from pptx.util import Inches, Cm
import os

# Define the directory containing PNG files and the save path
directory_path = " PDF の保存フォルダ名 "
save_path = os.path.join(directory_path, "DB_presentation.pptx")

# Create a presentation
presentation = Presentation()

# Iterate over all PNG files in the directory
for file_name in os.listdir(directory_path):
    if file_name.lower().endswith(".png"):
        # Add a slide with a blank layout
        slide_layout = presentation.slide_layouts[6] # Blank layout
        slide = presentation.slides.add_slide(slide_layout)

        # Full path to the image file
        image_path = os.path.join(directory_path, file_name)

        # Get slide dimensions
        slide_width = presentation.slide_width
        slide_height = presentation.slide_height

        # Add the image to cover the entire slide
        slide.shapes.add_picture(image_path, 0, 0, width=slide_width, height=slide_height)

# Save the presentation
presentation.save(save_path)

print(f"PPTX file saved to {save_path}")
```

図4 画像ファイルの pptx 化

```
import os
from pptx import Presentation
import win32com.client
import azure.cognitiveservices.speech as speechsdk

# 入力ファイルと出力ディレクトリの設定
input_path = " pptx の保存先フォルダ名とファイル名 " # PPTX ファイルのパス
output_dir = " pptx の保存先フォルダ名 " # 出力ディレクトリ
os.makedirs(output_dir, exist_ok=True)

# Azure Speech service configuration
speech_key = " Azure Speech API key " # Replace with your Azure Speech API key
service_region = "japaneast" # Replace with your Azure region
speech_config = speechsdk.SpeechConfig(subscription=speech_key, region=service_region)

# スライドを画像化
def slide_to_images():
    print("PowerPoint を開き、スライドを一括エクスポートします...")
    # PowerPoint を開く
    powerpoint = win32com.client.Dispatch("PowerPoint.Application")
    powerpoint.Visible = 1
    presentation = powerpoint.Presentations.Open(input_path, WithWindow=False)

    try:
        # すべてのスライドをエクスポート
        for slide_num in range(1, presentation.Slides.Count + 1):
            print(f"スライド番号 {slide_num} をエクスポートしています...")
            slide = presentation.Slides(slide_num)
            image_path = os.path.join(output_dir, f"slide_{slide_num}.png")
            slide.Export(image_path, "PNG")
            if not os.path.exists(image_path):
                print(f"エクスポートに失敗しました: {image_path}")
            else:
                print(f"エクスポート成功: {image_path}")

    except Exception as e:
        print(f"スライドエクスポート中にエラーが発生しました: {e}")

    finally:
        # プレゼンテーションを閉じる
        presentation.Close()
        powerpoint.Quit()

# ノートを Azure で音声に変換
def notes_to_audio(slide, slide_num):
    print(f"スライド番号 {slide_num} のノートを音声化しています...")

    notes_slide = slide.notes_slide
    notes_text = notes_slide.notes_text_frame.text if notes_slide else "ノートがありません。"
    audio_path = os.path.join(output_dir, f"slide_{slide_num}.wav")
```

```
try:
    audio_config = speechsdk.audio.AudioOutputConfig(filename=audio_path)
    speech_synthesizer = speechsdk.SpeechSynthesizer(speech_config=speech_config,
audio_config=audio_config)
    result = speech_synthesizer.speak_text_async(notes_text).get()

    if result.reason != speechsdk.ResultReason.SynthesizingAudioCompleted:
        raise Exception(f"音声合成に失敗しました: {result.reason}")

except Exception as e:
    print(f"音声化中にエラーが発生しました: {e}")
    audio_path = None

return audio_path

# 動画を作成
def create_video(image_path, audio_path, output_video_path, duration=5):
    print(f"動画を作成しています: {output_video_path}...")
    final_output_path = output_video_path.replace(".mp4", "_final.mp4")

    try:
        os.system(f"ffmpeg -y -loop 1 -i {image_path} -i {audio_path} -c:v libx264 -t
{duration} -pix_fmt yuv420p {output_video_path}")
        os.system(f"ffmpeg -y -i {output_video_path} -c:v copy -c:a aac {final_output_path}")
        os.remove(output_video_path)
    except Exception as e:
        print(f"動画作成中にエラーが発生しました: {e}")
        final_output_path = None

    return final_output_path

# 処理の流れ
slide_to_images()
presentation = Presentation(input_path)
videos = []

for i, slide in enumerate(presentation.slides, start=1):
    print(f"現在 {i}/{len(presentation.slides)} スライドを処理中です...")
    try:
        image_path = os.path.join(output_dir, f"slide_{i}.png")
        if not os.path.exists(image_path):
            print(f"スライド {i} の画像が見つかりません。スキップします。")
            continue

        audio_path = notes_to_audio(slide, i)
        if not audio_path:
            print(f"スライド {i} の音声生成に失敗しました。スキップします。")
            continue
```

```

video_path = os.path.join(output_dir, f"slide_{i}.mp4")
final_video_path = create_video(image_path, audio_path, video_path)
if final_video_path:
    videos.append(final_video_path)

except Exception as e:
    print(f"スライド {i} の処理中にエラーが発生しました: {e}")

print(f"作成された動画: {videos}")

```

図5 音声付き動画の作成

```

import os
import comtypes.client
import cv2
import ffmpeg
import azure.cognitiveservices.speech as speechsdk
from pptx import Presentation

# Paths
pptx_path = os.path.abspath('PDFの保存先フォルダ名とPDFファイル名') # Use absolute path
output_dir = os.path.abspath('PDFの保存先フォルダ名')

# Azure Speech service configuration
azure_speech_key = "Azure Speech API key" # Replace with your Azure Speech API key
azure_service_region = "japaneast" # Replace with your Azure region

# Function to generate audio using Azure TTS
def generate_audio(text, audio_path):
    speech_config = speechsdk.SpeechConfig(subscription=azure_speech_key,
    region=azure_service_region)
    speech_config.speech_synthesis_language = "km-KH" # Change language to Khmer
    speech_config.speech_synthesis_voice_name = "km-KH-PisethNeural" # Male Khmer voice
    audio_config = speechsdk.audio.AudioOutputConfig(filename=audio_path)

    synthesizer = speechsdk.SpeechSynthesizer(speech_config=speech_config,
    audio_config=audio_config)
    result = synthesizer.speak_text_async(text).get()
    if result.reason != speechsdk.ResultReason.SynthesizingAudioCompleted:
        raise Exception(f"Speech synthesis failed: {result.reason}")

```

図6 クメール語での音声作成

(参考2) 公的統計集計や分析に用いるプログラム言語の選定

非英語圏の国において、SAS を公的統計の集計や分析に用いるプログラム言語としての推進を図るうえで、表3と表4に示す事項の検討を実施した。

なお、このプログラム言語の選定においては、SAS や STATA は公的統計における信頼性の高いプログラム言語であり、特に政府機関や研究機関で多く利用されている。これらの言語・環境を考慮したうえで、以下の3つを推奨言語として提案している。

表3 プログラム言語の比較表

言語名	信頼性	バージョン更新の継続性	完全性（数値計算の精度・安定性）	選定理由
SAS	非常に高い	非常に高い（商用サポートがあり、長期サポートがある）	高い（商用ソフトウェアのため計算結果が安定）	政府機関で広く使用されており、長期間のサポートがある。データ管理や統計分析に強い。
Fortran	非常に高い	高い（古いコードも互換性が高い）	非常に高い（IEEE 754 準拠で高精度計算が可能）	科学技術計算向けに最適化され、CPI のような長期的な統計計算に適している。
STATA	高い	高い（バージョン間の互換性が比較的良好）	高い（浮動小数点処理が安定）	統計解析に特化し、長期的な安定性がある。バージョン間の互換性も比較的高い。

表4 Python や R の懸念点

言語名	懸念理由
Python	Python のバージョン更新は頻繁であり、例えば NumPy や pandas の API 変更が発生する。標準の浮動小数点演算（float）は IEEE 754 準拠だが、デフォルトでは 16 桁以上の計算精度が保証されない（decimal.Decimal を利用すれば回避可能）。商用サポートがないため、公的統計の長期運用には不安がある。
R	CRAN のパッケージの更新頻度が高く、バージョン間の非互換性が発生する。R 3.x 系から 4.x 系への移行時に関数の挙動が変更された事例があり、公的統計の長期運用にはリスクがある。高精度な計算は可能だが、バージョン固定（Docker などのコンテナ管理が必要）が必須である。

選定におけるまとめ

SAS は政府機関や公的統計の標準ツールとして広く使用され、商用サポートがあるため長期利用に適している。Fortran は科学技術計算の標準であり、CPI 集計に最適である。STATA は経済統計や公的統計向けに最適化されており、長期的なデータ管理に強い。一方、Python や R はオープンソースであるため、商用サポートがなく、バージョン間の互換性や計算精度の保証が課題となる。以上の点から、公的統計の集計用途には SAS, Fortran, STATA の3つが最適である。

データ可視化やライブラリの充実度では、オープンソースの Python や R が優位であるが、図7のように連携して用いることでそれらの点をカバーすることが可能である。

```
import saspy

# SAS セッションを開始
sas = saspy.SASsession()

# SAS コードの実行
sas_code = """
data sample;
  input x y;
  datalines;
1 2
3 4
5 6
;
run;

proc print data=sample;
run;
"""

# SAS コードを実行して結果を取得
results = sas.submit(sas_code)
```

図7 Python からの SAS コードの実行例

参 考 文 献

1. 李, 相穆 (2019). 外国語教育における視覚情報の効果. 言語文化論究, 42, 59-64. 九州大学大学院言語文化研究院.
2. 中池, 宏行 (2018). 学習方法の違いによる語彙習得率の比較研究. 千葉県立匝瑳高等学校, 第16回研究助成C, 調査部門・報告I.
3. 成田, 真澄 (2018). 日本人英語学習者の英語ライティングを伸ばす指導の方向性. Learner Corpus Studies in Asia and the World, 3, 43-60..
4. OpenAI. (2025). ChatGPT. Retrieved from <https://openai.com/chatgpt>
5. Canva. (2025). Canva. Retrieved from <https://www.canva.com>
6. Google. (2025). NotebookLM. Retrieved from <https://notebooklm.google>
7. Google. (2025). Gemini. Retrieved from
8. <https://www.nis.gov.kh>
9. <https://www.microsoft.com/ja-jp/windows/windows-11-specifications>
10. SAS Institute (2022). SAS® 9.4 Language Reference: Concepts.
11. Bureau of Labor Statistics (BLS), U.S. Department of Labor: CPI Methodology.
12. Metcalf, M., Reid, J., & Cohen, M. (2018). Modern Fortran Explained. Oxford University Press.
13. High Performance Computing Modernization Program (HPCMP), U.S. Department of Defense.
14. StataCorp LLC (2023). STATA User's Guide.
15. International Monetary Fund (IMF) Data & Statistics Manual.
16. Python Software Foundation: Python 3. x Documentation.
17. NumPy User Guide.
18. R Core Team (2023). The R Language Definition.
19. Wickham, H. (2019). Advanced R. CRC Press.