

〈PoKeBo Studio〉 構成主義的学習に向けた プレイフルインタフェースの提案

多田 魁登^{*1} 長谷川 孔明^{*1} 大島 直樹^{*2} 岡田 美智男^{*1}

PoKeBo Studio: Proposal for a playful interface for constructivist learning

Kaito Tada^{*1}, Komei Hasegawa^{*1}, Naoki Ohshima^{*2} and Michio Okada^{*1}

Abstract -Methods for learning programming and logical thinking while designing the behavior of robots and agents are becoming increasingly popular. In addition to learning programming, is it possible to expand the use of this method to writing and scenario creation? In our presentation, we will introduce the basic concept and prototype of PoKeBo Studio, a playful interface for learning text creation and scenario creation in a constructivist way through visual programming of conversations among multiple robots.

Keywords: Constructivism, Playful interfaces for writing and scenario creation and Interaction Design

1. はじめに

繰り返し遊んでいるうちに、知らず知らず能力が伸びていく。そんな経験はないだろうか。パズルやクイズ、スポーツなど、伸びる能力は違うが楽しみながら能力を伸ばすものはいくつも存在する。例えば、ロボットを制御するために試行錯誤しながらプログラミングを行うことで、論理的思考が鍛えられる。

一方で、2020年度から小学校でのプログラミング教育が必修化され、論理的思考の育成が教育に取り入れられた。ただし、あくまでも論理的思考の育成が狙いで、プログラミング言語を覚えたり、プログラミングの技能を習得したりすること自体を狙ってはいない。また、数学や理科、社会など様々な教科にプログラミングを取り入れて実施することが期待されている。

我々は、文章作成の新たな形として、遊び感覚でプログラミングしながら文章作成やシナリオ作成といった能力を知らず知らずのうちに鍛えるプレイフルインタフェース〈PoKeBo Studio〉を構築した。本稿では、その基本コンセプトとインタラクションについて述べる。

2. 研究背景

2.1 授業形式の変化

学校の授業では、教員が教科書や資料の情報を黒板やホワイトボードなどを用いて生徒達に説明し、生徒達は一方的に聞くだけの受動的な形式が多い。これは認知主義的学習に当てはまる。また、漢字の書き取りや計算ドリルを何度も繰り返し解くなどといった、課題として提示されることが多い学習方法は行動主義的学習に当ては

まる^[1]。

対して、近年注目されている学習方法がアクティブラーニングやオルタナティブ教育である。これらは生徒達が討論したり、興味のある事柄について支えてもらいながら挑戦する学習方法であり、構成主義的学習に当てはまる。

2.2 構成主義的学習

「構成主義」とは認識論の一つで、「知ることとは自分の中に意味を構成すること」と考える立場である。「構成主義」では、学びが起こる瞬間は教育者から学習者に知識が付与されたときではない。そもそも知識は客体として存在するのではなく、学習者が構成するものである。すなわち、学習者が自分の言葉で理解したり、腑に落ちたと感じられたり、自分一人で実践できるようになったりするときに学びが起こると考える^[2]。

2.3 文章作成の方法

小学生が文章を書く場面として最も多いのが課されたテーマについて文章を書くことである。こういった場面での文章作成は、宛て名がはっきりしないまま書くということが起きてしまう。大人が文章を書く際、誰に向けて、何のためにといった、意図や目的がはっきりしたうえで文章を書く。しかし、小学生が課されたテーマに対して作る文章はただ言葉を並べただけになってしまう可能性がある。そのため、社会に出た際の文章作成の能力を養うのには不十分であるといえる。

本研究では、ロボットたちのおしゃべりを遊び感覚で作成することで、知らず知らずのうちに文章作成やシナリオ作成能力を養えるシステムの実現を狙いとする。

3. 〈PoKeBo Studio〉のコンセプト

ロボットの動作をデザインしながら、その過程でプログラミングや論理的思考を学ぶ手法が普及しつつある。その手法を文章作成やシナリオ作成などに活用の幅を広

*1: 豊橋技術科学大学 情報・知能工学系

*2: 豊橋技術科学大学 エレクトロニクス先端融合研究所

*1: Department of Computer Science and Engineering,
Toyohashi University of Technology

*2: Electronics-Inspired Interdisciplinary Research Institute,
Toyohashi University of Technology.

げられないか、という考えから生まれたのが〈PoKeBo Studio〉である。

〈PoKeBo Studio〉はインタフェースに Scratch を、ロボットに〈PoKeBo Cube〉を使用する、プレイフルインタフェースである。Scratch を用いてプログラミングし、〈PoKeBo Cube〉の動作やおしゃべりの内容を設計する。論理的思考や文章の作成方法といったことをあまり意識させず、遊び感覚で試行錯誤を繰り返しているうちにそれらの能力を身に着けられることがコンセプトである。

4. 〈PoKeBo Studio〉の構成

4.1 システム構成

〈PoKeBo Studio〉のシステム構成を図1に示す。本システムは〈PoKeBo Cube〉と Web サーバから構成される。

Web サーバには Node.js を使用し、Scratch のサービスを提供している。Scratch で作ったプログラムは Web サーバで解析し、Robot Operating System (ROS)を用いて〈PoKeBo Cube〉へ送信される。

〈PoKeBo Cube〉では、受け取ったプログラムを元に頭を動かしながらおしゃべりを行う。

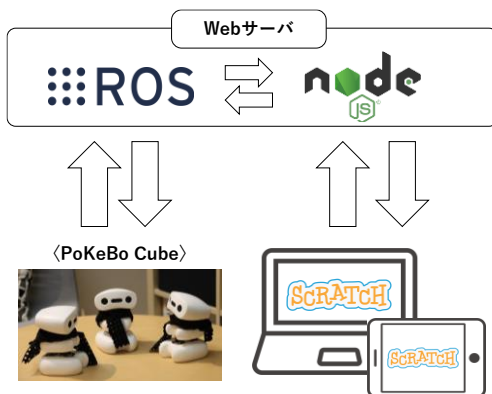


図1 〈PoKeBo Studio〉のシステム構成
Fig.1 System Configuration of “PoKeBo Studio”.

4.2 〈PoKeBo Cube〉

〈PoKeBo Cube〉とは情報提供の仕方をタンジブルに調整できるソーシャルインタフェースとして開発されたロボットである^[3]。3体のロボットによるクローズドモードとオープンモード形式の会話を、ロボット達の向きや位置を変えることで切り替えることができる。



図2 〈PoKeBo Cube〉の外観
Fig.2 Appearance of “PoKeBo Cube”.

本システムでは、〈PoKeBo Cube〉を用いて入力した文章による会話の場を生成する。

4.3 Scratch

Scratch (スクラッチ) は、アメリカ・マサチューセッツ工科大学のメディアラボが無償で公開しているビジュアルプログラミング言語である。画面上のブロックをつなぎ合わせてプログラムを作成する。8歳から16歳向けにデザインされており、200ヶ国以上の国と地域で利用され、70以上の言語で使用可能である^[4]。また、ブロックを追加することで Scratch のプログラムを拡張することが可能である。

本システムでは、Scratch にブロックを追加してプログラムの拡張を行い、文章作成・ロボット制御のインタフェースとして利用する。

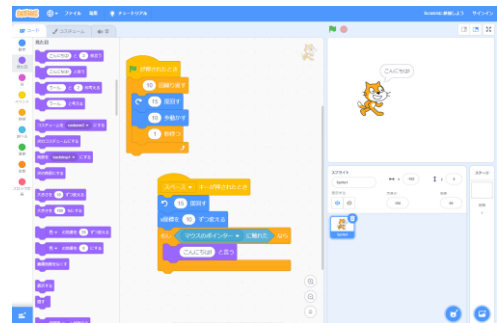


図3 Scratchの外観
Fig.3 Appearance of “Scratch”

4.4 ハードウェア構成

本システムの一部である〈PoKeBo Cube〉のハードウェア構成を図4に示す。

〈PoKeBo Cube〉には DC モータが2個とそれを制御するための制御回路、外装の最下部にはソニー社の toio が組み込まれている。

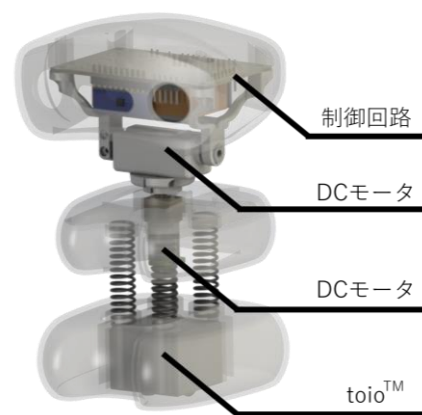


図4 〈PoKeBo Cube〉のハードウェア構成
Fig.4 Hardware configuration of “PoKeBo Cube”.

4.5 ソフトウェア構成

〈PoKeBo Studio〉では Robot Operating System(ROS)と Node.js を用いて〈PoKeBo Cube〉の制御と Scratch の提

供を行っている。ソフトウェア構成図を図5に示す。

Node.js では Scratch 用の Web サーバを立てており、ROS 部分と通信するための ROS のノードも組み込んでいる。

ROS 内では、Scratch から送られてきたデータを処理するノードや〈PoKeBo Cube〉を制御するノードを複数用意している。

入出力ノードは、Scratch と〈PoKeBo Cube〉を繋ぐ役割を担っている。Scratch 内で入力された情報を指令ノードへ送信、もしくは指令ノードからのフィードバックを Scratch へ返す。

指令ノードでは、受け取った命令を〈PoKeBo Cube〉に適した命令の形へ変換し、サーボ制御ノードと発話ノードへと送信する。また、実行結果や〈PoKeBo Cube〉の状態などのフィードバックを入出力ノードへ送信する。

サーボ制御ノードは、指令ノードから送信された DC モータの角度・速度情報または右を向く、頷くなどの基本動作情報に従い、DC モータを動かす。

発話ノードでは、指令ノードから送信されたテキスト情報を合成音声によって音声データに変換しスピーカから再生する。合成音声には ATR-Promotions 社の WizardVoiceSDK を用いており、話者データベースには「女の子」を指定している。

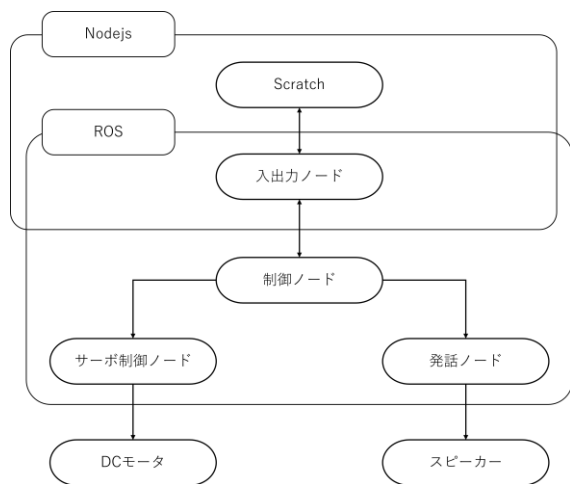


図5 〈PoKeBo Studio〉のソフトウェア構成
Fig.5 Software configuration of “PoKeBo Studio”.

5. インタクションデザイン

〈PoKeBo Studio〉は Scratch を入出力のインタフェースとして、〈PoKeBo Cube〉を出力のインタフェースとして使用する。

インタクションとしては以下のような流れを想定している。

- ① Scratch のブロックに〈PoKeBo Cube〉の動作や文章を入力する。
- ② Scratch を実行し、命令を〈PoKeBo Cube〉へ送信する。

③ つなぎ合わされたブロックに合わせて〈PoKeBo Cube〉が会話や仕草を行う。

④ 〈PoKeBo Cube〉の仕草や会話から、プログラムのフィードバックを行う。

⑤ 入力した文章やプログラムを修正する。

⑥ 再度実行し、フィードバックを行う。

6. おわりに

本稿では、タンジブルなインタフェースで情報を教えてくれる〈PoKeBo Cube〉と、直感的な操作でプログラミングができる Scratch を組み合わせた、作文教育のための構成主義的学習に向けたプレイフルインタフェース〈PoKeBo Studio〉の基本コンセプトとプロトタイプについて紹介した。

今後は、実際に使用した際の印象の調査を通してブラッシュアップを行い、より使いやすいシステムへ改善を行う。また、作文教育へ導入できるシステムに向けて、実験等を通して、より良いインタラクションの形を見つけていく。

参考文献

- [1] 西城 卓也: 行動主義から構成主義, 医学教育, 43 巻, 4 号, 290-291(2012)
- [2] 久保田 賢一: 構成主義が投げかける新しい教育, コンピュータ&エデュケーション, 2003, 15 巻, 12-18(2014)
- [3] 真弓 凌輔, 上野 慎介, 大島 直樹, 岡田 美智男: PoKeBo³ 共構築型コミュニケーション空間の提案, ヒューマンインタフェースサイバーコロキウム, 67-70 (2020)
- [4] Massachusetts Institute of Technology, National Science Foundation, Siegel Family Endowment, LEGO Foundation: Scratch について, Scratch, <https://scratch.mit.edu/about> (2022)