日本語環境ブロックプログラミングと連携した ソースコードの穴埋め選択問題生成システム

A System for Creating Fill-in-the-blanks Multiple Choice Questions of Source Code for Block Programming Environment in Japanese

島岡 慎也，鷹野 孝典
Shinya SHIMAOKA，Kosuke TAKANO
神奈川工科大学情報学部情報工学科
Department of Information and Computer Sciences, Faculty of Information Technology, Kanagawa Institute of Technology
Email:s1821121@cco.kanagawa-it.ac.jp

あらまし：本稿では，ブロックプログラミングと連携したソースコードの穴埋め問題生成システムを提案する．提案システムは，穴埋め問題生成処理において，ワンパターンの出題を生成しないように，選択肢の類似性を調整して選択肢の組み合わせを決定する点が特徴である．プロトタイプを用いた実験により，提案する選択肢穴埋め問題生成手法により，適切に問題生成できるかを評価する．

キーワード：ブロックプログラミング，問題生成，穴埋め，選択肢，コーディング力

1. はじめに

プログラミング学習において，ブロックプログラミングにより論理的思考を養うことできる．一方，システム開発では，コーディング力が必要とされる．本研究では，プログラミング学習者が，ブロックプログラミングを用いることで論理的思考力を身に着けるとともに，ブロックプログラミングに対応するソースコードを利用してコーディング力を養成するシステムの実現を目標とする．本研究では，この目標のためにブロックプログラミングと連携したソースコードの穴埋め問題生成システムを提案する．提案システムは，穴埋め問題生成処理において，ワンパターンの出題を生成しないように，選択肢の類似性を調整して選択肢の組み合わせを決定する点が特徴である．

2. 提案システム

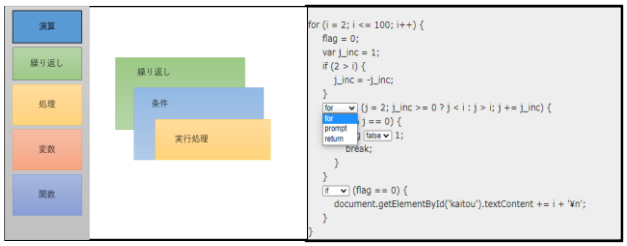

図 1 提案システムのユーザインタフェース

2.1 概要

図 1 は提案システムのユーザインタフェース例を示している．図 1 では，左側に日本語ブロックプログラミング環境，右側にブロックに対応したソースコードが表示されている．日本語ブロックプログラミング環境を用いる理由は，日本語を母国語とするプログラミング初級者に対して，論理的思考を養うのに適しているからである．ソースコードは，穴埋め選択問題となっている．学習者は，ブロックプログラミング後に，穴埋め選択問題を解くことで，論理的に考えながら，少しずつコーディング力を高めることができる．

2.2 穴埋め問題生成手法

穴埋め問題生成手法では，ブロックプログラミング環境で利用できる予約語や演算式記号などの要素を，要素が持つ意味や演算機能の類似性などを基準にしてグループ化(表 1)しておき，ソースコード中に含まれる要素を穴埋め箇所とした場合に，グループ中の要素を選択肢として選択問題を生成する．本手法の特徴は，ソースコード中の穴埋め箇所数を変化させるだけでなく，異なるグループの要素を組み合わせることで，ワンパターンの出題にならないように難易度を調整して穴埋め選択肢問題を生成する点にある．

表 1 選択肢グループ例

グループ	要素
G 1	'for', 'while', 'do'
G 2	'if', 'else', 'switch'

G 3	'break', 'continue'

3. 実験

提案する選択肢穴埋め問題生成手法により，適切に問題生成できるかを評価する．実験に利用した問題数は，全体で 354 問である．難易度は，簡単な問題と難しい問題が 89 問ずつ，普通の問題が 176 問である．表 1 に評価基準を示す．

なお，提案システムのプロトタイプの構築は，HTML/CSS および JavaScript 言語を用いて，Web システムとして実装した．ブロックプログラミングのユーザインタフェースの導入には Google が提供する Google Blockly ライブラリを用いた．

表 1 の評価基準で評価した結果を図 2 に示す．図 2 で，基準 4 の結果は分散の値となっている．また，実験で実際に生成された問題の例を図 3 と図 4 に示す．図 3 では，同一グループから選択肢を生成しており，実際に生成されている選択肢はいずれも条件分岐のための予約語となっており，類似性が確認できる．図 4 では，異なるグループから選択肢を生成している．学習が進むと，前後に利用されている語との比較によっても解答の導出が可能である．

図 2 の評価，および図 3 と図 4 の生成例より，異なるグループの要素を組み合わせることで，ワンパターンの出題にならないように難易度を調整して穴埋め選択肢問題を生成できることが確認できる．

表 1 評価基準

	基準	評価
1	正解となる解答が選択肢に入っている．	Yes / No
2	設問内で選択肢が複数かぶっていない．	Yes / No
3	ブロックプログラミングより解答が推測できる．	Yes / No
4	出題内容の種類	予約語 不等号 四則演算 その他
5	選択肢のみから正解が推測できない．	Yes / No

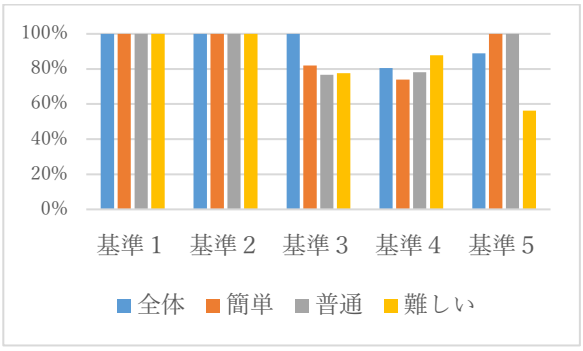

図 2 評価基準に基づいた評価結果

4. まとめと今後の課題

本研究では，ブロックプログラミングと連携したソースコードの穴埋め問題生成システムを提案した．実験結果より，提案するブロックプログラミングと連携したソースコードの穴埋め問題生成システムを実現する見込みを得ることができた．

今後の課題として，今回は事前に用意したキーワードから問題を生成したため，変数名を選択肢として利用することができない．このため，問題の生成方法の見直しが必要である．さらに，異なるグループにまたがって選択肢を生成する際に，選ばれるキーワードをランダムで抽出した．しかし，キーワードの出現確率に重みを設定するなど，初学者向けのキーワードがより多く出現させるなど，難易度の調整方法の工夫が必要である．提案システムを実用的なシステムとして実現するために，これらの課題について検討していきたい．

```
for (i = 2; i <= 100; i++) {
    flag = 0;
    var j_inc = 1;
    if (2 > i) {
        = -j_inc;
    for (j = 2; j_inc < 0 ? j < i ? j > i; j += j_inc) {
        if (i % j < 0) {
            flag = 1;
            break;
        }
    }
    if (flag == 0) {
        document.getElementById('kaitou').textContent += i + '¥n';
    }
```

図 3 同じグループから生成された問題例

```
var a, b, c, goukei;


a = 1;
b = 2;
c = 0;
goukei round a + b;
document.getElementById('kaitou').textContent += goukei + '¥n';
goukei ;
document.getElementById('kaitou').textContent += goukei + '¥n';
goukei split b + c;
document.getElementById('kaitou').textContent += goukei while '¥n';
```

図 4 異なるグループから生成された問題例

参考文献

(1) 内田保雄：初級プログラミングのための自動作問システム，情報処理学会研究報告コンピュータと教育（CE），Vol.2007，123(2007-CE-092)，pp.109-113 (2007).

(2) 野上裕二，納富一宏：プログラミング学習支援における問題自動生成に関する基礎的検討， 情報処理学会 第16回情報科学技術フォーラム(FIT2017)講演論文集，第 3 分冊，K-022, pp.465-466 (2017).

(3) E. Pasternak, R. Fenichel and A. N. Marshall, "Tips for creating a block language with blockly," 2017 IEEE Blocks and Beyond Workshop (B&B), pp. 21-24 (2017).