Processingアプリケーション開発のための視覚的ドメイン特化言語

栗原 あずさ　佐々木 晃　脇田 建　細部 博史 *

概要. 本稿ではアートやデザインのためのプログラミング言語である Processing をビジュアルブロックによって記述するドメイン特化言語 (以下 DSL) の開発について述べる．Processing は視覚的な表現を平易なコードで実現できるため，高校などのプログラミング教育に利用される．しかし初学者はプログラミングの理解にコストがかかる．そこでブロックベース DSL を用いて Processing のコードを記述する．ブロックは自然言語で表現され形状で視覚的に区別されるため可読性が高い．ブロックの組み合わせにより視覚的・直感的なコーディングが可能になりプログラミング習得のコストを削減する．DSL は WEB アプリケーションとして提供されるため導入のコストが少なく，ライブコーディングによりデバッグを容易にした．

1 はじめに

Processing [2] はアートやデザインのためのプログラミング言語である．これは簡潔な表現で視覚的な結果を得られるため，デザインだけでなく高校などでプログラミング教育に利用されている．こういったプログラミング初学者がコードを書く際の問題に (1) プログラミングのコンセプト（順次実行，分岐，反復）が理解できない，(2) 関数や命令が理解できない，(3) デバッグができない，などがある．

本研究では Processing のコードを視覚的なブロックで記述するドメイン特化言語（以下 DSL）を作成した [6]．ユーザはブロックの直感的な組合わせでプログラミングする．ブロックとその組合わせ方は文法的に正しいため文法エラーは発生せず，文字入力はパラメータのみであるためタイプミスが少ない．これを用いて前述の問題を (1) ブロックの形状でフローを可視化する，(2) ブロックのはたらきを日本語で記述する，(3) ライブコーディングにより実行結果の確認を容易にする，という方法で解決を図る．

本 DSL はブロックの追加と削除によって拡張する．初学者には不要な機能を制限することで DSL を単純化し，コードのパターンである tips ブロックを追加することで初学者の利用に特化した Processing の拡張を実現した．初学者が視覚的 DSL を通じて Processing のしくみとプログラムのコンセプトを理解することで，テキストベースの Processing や他のプログラミング言語への移行を容易にする．

2 Processing

Processing は通常 Processing Development Environment と呼ばれる IDE やテキストエディタである Sublime Text 上でプログラミングを行う．ここではデバッガ，入力補完，エラーチェック，実行中のパラメータ変更などが利用可能であるが，初学者はこれらの機能を効果的に利用できない．

Processing は Java の API として実装されており static，active，Java の三つのプログラミングモードがある．static は静止画，active は動画の描画に利用される．Java モードはクラスなどの機能が利用できる．本 DSL は active モードに固定することで機能を限定しプログラムの構造を単純にする．

3 提案手法

3.1 視覚的 DSL の設計

本 DSL はブロックベースの視覚的言語から Processing のコードを生成する外部 DSL であり，内部 DSL として Processing.js を用いブラウザ上で Processing を実行する．

既存の子供向けプログラミング学習ツールには Scratch [3]，blockly [1]，Viscuit[4] などがある．Scratch と blockly はブロックベースの言語である．ブロックは高度な命令を平易な日本語で表現するため，英語が苦手なユーザも読み書きしやすく命令や関数のはたらきを明確に記述できる．またブロックで記述することで構文エラーや型エラーが発生しない．

本 DSL で記述するプログラムは以下のテンプレートの形に変換される．

```
// 変数定義
void setup(){
  // setup の本体
}
void draw(){
  frameRate(n);
  // draw の本体
}
// イベントハンドラなどの関数
```


* Azusa Kurihara, 法政大学大学院情報科学研究科，Akira Sasaki and Hiroshi Hosobe, 法政大学情報科学部，Ken Wakita, 東京工業大学大学院情報理工学研究科

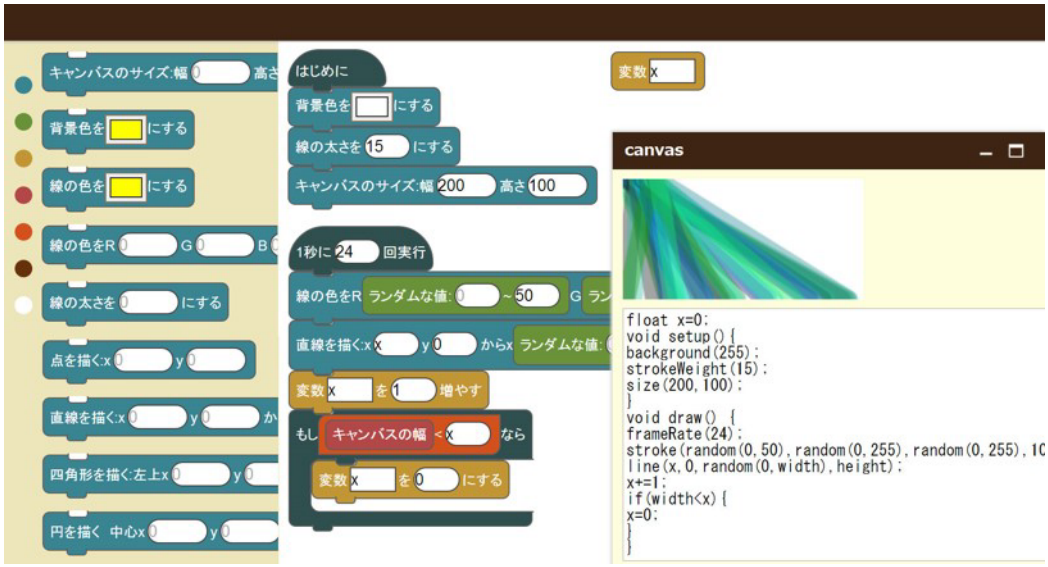

図 1. DSL の外観．左から，パレット，ワークスペース，ポップアップ（キャンバスとプログラム）．

3.2 視覚的 DSL の実装

本 DSL は WEB アプリケーションとして提供されるため導入が容易である．DSL にはブロックのプログラミング環境，ブロックを Processing のコードに変換するトランスレータ，ブラウザで Processing を実行する Processing.js が含まれている．本 DSL はプログラムに変更があるたび変換および実行するライブコーディングを実現した．

ブロックは色と形状，日本語のラベル，変換規則によって定義される．ブロックは一行以上の関数や変数などに変換される．例えば「n 回繰り返す」ブロックは次のコードに変換される．

```
for(int i=0;i<n;i++){
  // for 文の内容
}
```

このように，パラメータをあらかじめ決めることで具体的なブロックを定義し，ブロックのはたらきを明示的に表現できる．本稿ではコードに頻出するパターンを表すブロックを「tips」と呼ぶ．tips を定義することでユーザがこれらを再定義する必要が無くなるため，全体のブロック数が減り可読性が高まる．

4 まとめと今後の課題

本稿では Processing およびプログラミングのコンセプトを理解するための初学者向け視覚的 DSL の設計と実装について述べた．本 DSL は Processing のコードに変換可能なブロックを用い，ライブコーディングによりデバッグを容易にする．一部機能を制限しパターンを tips ブロックにまとめることで初学者による記述に特化した DSL を作成した．

今後の課題には，より大規模かつ複雑なプログラムの記述と Processing 以外の言語での DSL の作成が挙げられる．プログラムの規模が大きくなり複雑化するにつれブロックの数は増加するが，可読性の高さを維持する必要がある．以前の研究 [5] では，マクロブロックを定義可能にすることでブロックの表現の幅を広げた．また，本 DSL はブロックの日本語表記と変換規則を定義し Processing 以外の言語のためのブロックを作成することで，より汎用的なブロックベースの言語の作成を支援する．

謝辞

本研究は JSPS 科研費 25540029 の助成を受けたものである．

参考文献

[1] N. Fraser, et al. blockly - A visual programming editor, 2012. https://code.google.com/p/blockly/.

[2] C. Reas and B. Fry. Processing.org, 2001. https://www.processing.org/.

[3] M. Resnick, et al. Scratch: Programming for All. *Communications of the ACM*, 52(11):60–67, 2009.

[4] 原田康徳, 加藤美由紀, R. Potter. Viscuit: 柔軟な動作をするビジュアル言語. WISS2003 論文集, pp. 47–56, 2003.

[5] 栗原あずさ, 佐々木晃, 脇田建. ビジュアルブロックを採用したドメイン特化言語とその開発ツールの実現手法. 信学技報, 113(489):97–102, 2014.

[6] 栗原あずさ, 佐々木晃, 脇田建, 細部博史. Processing アプリケーション開発のための視覚的ドメイン特化言語の実装. 日本ソフトウェア科学会第 31 回大会講演論文集, 2014.