

Web ベースグラフィカルプログラミングエディタを用いた Flex プログラミング環境の開発

Development of a Flex Programming Environment using a Web-based Graphical Programming Editor

尾崎 陽一^{*1}, 香川 考司^{*2}
Yoichi OZAKI^{*1}, Koji KAGAWA^{*2}

^{*1} 香川大学大学院工学研究科

^{*1} Graduate School of Engineering, Kagawa University

^{*2} 香川大学工学部

^{*2} Faculty of Engineering, Kagawa University

Email: s13g454@stmail.eng.kagawa-u.ac.jp

あらまし: 著者らの所属する学科で開講されているコンパイラに関する演習では、C 言語の字句解析器生成系である Flex が用いられる。Flex は文法的に複雑な言語であり、コンパイラの学習と Flex の文法を同時に学ぶことは学習者にとって大きな負担となる。本研究では Web ベースグラフィカルプログラミングエディタ Blockly を用いて文法を意識せずに行える Flex プログラミング環境の開発を行う。Web ベースで導入が容易である。また、リアルタイムのソースコード出力により、Flex の文法学習にスムーズに移行することができる。

キーワード: プログラミング学習, Flex, Blockly, Web ベース

1. はじめに

著者らの所属する学科で開講されているコンパイラに関する演習では、字句解析の演習に Flex が用いられる。Flex は、正規表現とそれに対する動作記述から、C 言語の字句解析プログラムを自動生成する字句解析器生成系である。学習者はコンパイラの学習に加え、Flex の文法も学ぶ必要がある。Flex の文法は複雑であり、同時にこの二つを学ぶことは、学習者にとって大きな負担となってしまう。そこで、文法を意識せずに Flex プログラミングが行えるシステムが必要となる。

文法を意識せずにプログラミングが行える環境として Scratch⁽¹⁾ が存在するが、現在 Scratch は Flex に対応していない。

空田の研究⁽²⁾では、グラフィカルパーツによる文法を意識せずに行えるプログラミング方法と、パーツと C 言語のソースコードの相互変換によって、プログラミングの基礎概念と、文法を切り離して学習できる支援環境の開発を行った。

中山の研究⁽³⁾では、動的なハイライト表示を行うエディタと、正規表現支援ツールによる Flex/Bison プログラミングの支援環境の開発を行った。

空田の研究では、グラフィカルパーツと C 言語のソースコードの相互変換を行うことができる。JSON で記述を行なうことにより他言語に対応可能であるが、制御構造が大きく異なる言語に対しては ActionScript で記述する必要があるため、学習者が Flex/Bison の文法をある程度知らないと有効に使う

ことができない。

本研究では、これらの研究の問題を解決するために、Web ベースグラフィカルプログラミングエディタ Blockly を用いて文法を意識せずに行える Flex プログラミング環境の開発を行う。

2. Blockly

Blockly (<http://code.google.com/p/blockly/>) とは、Google で開発されているグラフィカルなプログラミングエディタである。図 1 のようなブロックを繋ぎ合わせることでプログラミングを行う。このため構文エラーに悩まされず、直感的にプログラミングをすることができる。

全てのコンポーネントが JavaScript で記述されており、ドキュメントも豊富に用意されているため拡張性が高く、容易にカスタマイズを行うことができる。また、Blockly は Web ベースのアプリケーションであるため、学習者は導入の作業が不要である。さらに、Blockly で作成したプログラムは、JavaScript など別の言語のコードに変換してリアルタイムに自動生成される。現在は JavaScript, Python の出力をサポートしている。ユーザは自分の作成したプログラムのソースコードを常に確認できるため、他の言語への移行性が高い。

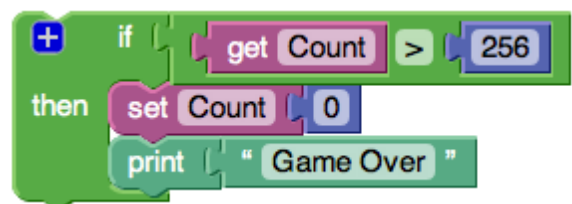


図1 Blockly のブロック

3. システムの実装

本研究では、Blockly で Flex プログラミングを行うために以下のような実装を行った。

3.1 ブロックの作成

Flex は、Blockly が本来サポートしている言語とは構造が大きく異なるため、Blockly 上で Flex プログラミングを行うためには新たなブロックを作成しなければならない。

主に以下のブロックを作成した

- 正規表現を記述するためのブロック
- 動作を記述するためのブロック
- 正規表現の定義を行うためのブロック
- それぞれのブロックを接続するベースとなるブロック

ユーザはこれらのブロックを組み合わせることで Flex プログラミングを行うことができる。

その他ユーザを支援する機能として、次の機能を実装した。

まず、ブロックにカーソルを合わせるとブロックの説明が日本語で表示される。正規表現を記述するブロックは、簡略化のためにブロック内の日本語の説明を省いたが、この機能によってユーザはブロックの使い方を容易に理解することができる。

また、文法的に誤ったブロックの接続が行われないうように型を定義した。例えば、本来動作を記述するブロックの接続が行われる個所には、正規表現を記述するブロックの接続を行うことはできない。

3.2 Flex コードの自動生成

Blockly にはユーザが繋ぎあわせたブロックからリアルタイムにソースコードを自動生成する機能がある。今回の研究では、この機能に Flex コードを自動生成できるように拡張した。また、Flex には文法上特別な意味を持つ特殊な文字が存在する。本システムでは、文字列中の特殊文字を判定し自動的にエスケープする機能を追加した。

Web ベースプログラミング環境構築プラットフォームである WappenLite⁽⁴⁾ との連携を行うことにより、Web ブラウザからプログラムが実行可能である。

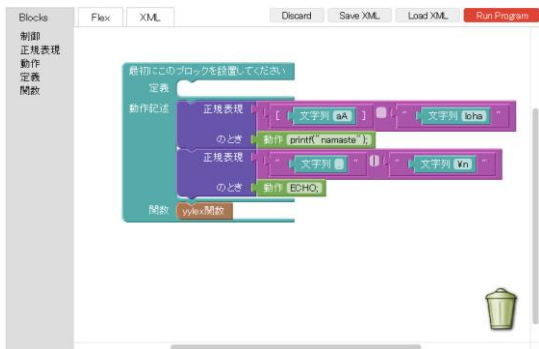


図2 作成したシステム

4. 評価

情報系学科の学生を対象に、本システムを使用してもらい感想を聞いた。その結果、良かった点として次のような意見が挙げられた。

- 直感的に操作方法が分かった
- 型の定義によって文法の間違いがあらかじめ分かるのが良い
- エディタでプログラミングを行うのと比べて分かりやすかった

これらの意見により、文法を意識せずに Flex プログラミングが行えると思われる。また、悪かった点として、次のような意見が挙げられた。

- 正規表現の接続を示すブロックや動作を記述するブロックが分かりにくい
- 接続部の形を変えるなどして接続できないブロックを区別する方法が欲しい
- 特殊文字の判定によって一部表現できない正規表現がある

これらの点は今後の課題として検討が必要である。

5. まとめ

コンパイラ学習者の支援のために、Web ベースグラフィカルプログラミングエディタ Blockly を用いて Flex プログラミング環境を開発した。Web ベースで導入が容易であり、学習者は文法を意識せずに Flex プログラミングを行うことができる。全てのコンポーネントは JavaScript で記述されており、容易にカスタマイズを行うことができる。

6. 今後の課題

評価の結果挙げられた問題に加え、次のような問題を検討している。

現在のシステムの仕様では、すでに接続されているブロックにさらにブロックを接続したい場合、ユーザが手動で接続部を追加しなくてはならない。より円滑な操作のためには、接続部を自動で追加する機能が必要である。

また、多くのブロックが接続されると横に長くなってしまい見づらくなってしまう。ブロックがある程度の長さになると改行を行う機能が必要である。

参考文献

- (1) John Maloney, Leo Burd, Yasmin Kafai, et al.: "Scratch: A Sneak Preview", C5 2004: Kyoto, Japan, pp. 104-109 (2004)
- (2) 空田卓也, 香川考司: "Adobe Flex によるグラフィカルなプログラミング学習支援環境", 教育システム情報学会研究報告 25(3), 香川大学幸町キャンパス, pp.3-6 (2010)
- (3) 中山和也, 香川考司: "AMF を利用したプログラミング学習支援システム", 教育システム情報学会研究報告 25(3), 香川大学幸町キャンパス, pp.7-12 (2010)
- (4) Koji Kagawa: "WappenLite: a Web Application Framework for Lightweight Programming Environments", ITHET 2010, pp.21-26 (2010)