ツリー表示を活用した Ajax ベースプログラミングエディタの構築

Development of an Ajax-Based Programming Editor Utilizing a Tree View

藤沢 尚樹[*1], 香川 考司[*2]
Naoki FUJISAWA[*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: s14g479@stmail.eng.kagawa-u.ac.jp

あらまし：Flex などで使われる正規表現は単純な文法に基づくが，初心者がその構文構造を正確に理解することは難しい．このため，ビジュアルプログラミング環境を用いて支援した．その結果，学習者は，構造を理解して目的の正規表現を作成できるようになった．しかし，入れ子が深くなるため、その操作は煩わしく、多くの表示スペースが必要になった．そこで，ブロックスタイルの利点を保ちながらも，より少ない操作と表示スペースを提供できる、ツリー表示を活用した Ajax ベースのプログラミングエディタを構築する．
キーワード：Ajax ベース，ブロックスタイル，入れ子構造，木構造

1. はじめに

Scratch [(1)] や Blockly (https://code.google.com/p/blockly/) などのブロックプログラミングエディタは, あらかじめ用意されたビジュアルパーツ (以下，単にパーツ）を組み合わせることでプログラミングするスタイルを提供する．これは，言語の文法の理解が不十分でも，目的のプログラムを容易に作成することを可能にしたものである．

そこで筆者らの研究グループでも, Blockly をベースに，大学生でも習得に苦労する Flex の正規表現の編集を支援する，Web ベースグラフィカルプログラミングエディタ（以下，先行システム）を開発した[(2)]．これは，既存のブロックスタイルエディタでは対象としていなかった，より複雑な式を持つ言語を支援する試みである．その結果，学習者自身が構造を理解して正規表現を作成できるようになった．しかし，入れ子の深い構造をブロックスタイルで扱った場合にはより多くのパーツを操作しなければならず，操作が煩わしくなり表示に多くのスペースを要するなどの問題があった．図 1-1 と図 1-2 に正規表現における例を示す．

```
"a"|"b""cd"*
"a"|"b"|"c""def"*
```

図 1-1　テキストで記述された正規表現

関数型言語 Haskell の場合，ブロックの連結方式に柔軟性がないとやはり表示スペースを取りすぎることが判明した[(3)]．

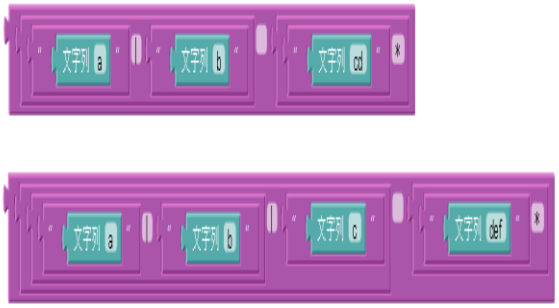

図 1-2　パーツで記述された正規表現

本研究の目標は，パーツの操作と表示スペースをより少なく抑え，ブロックスタイルのように複雑な構造を支援できるプログラミング環境を開発することである．

2. ツリー表示に基づく UI

先述の問題は, ブロックスタイルで深い入れ子構造を有する言語を支援した際に頻発する．これはブロックスタイルが構造を包含関係で指定することによる．このため本研究では，ツリー表示に基づくユーザインタフェース（以下，略して UI）を用いる．

正規表現は，連接と選択と反復の 3 つの演算に基づく．それらは図 2 のような木構造による表現が可能である．

そのようなツリー表示を利用した UI として，Cloud9 (https://c9.io/) や Eclipse のような統合開発環境でも既に利用されている，アウトラインエディタが挙げられる．Eclipse の場合，アウトラインエディタはクラスやメソッド等の階層を編集するために利用されているが，本研究では式の階層を編集するた

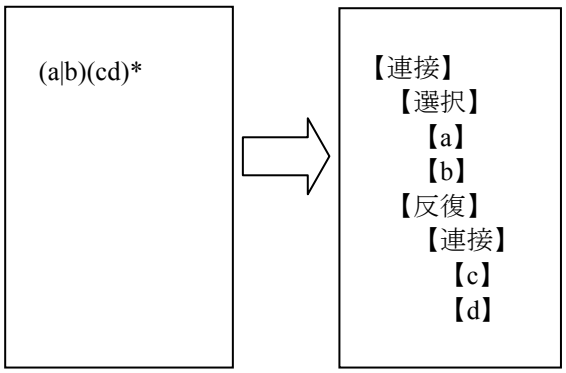

図 2　木構造で表現された正規表現

めに用いる．

木構造に基づく UI に求められる最低限の編集機能は，作成，入力，移動，削除などである．それを Web ベースで提供しているライブラリに jsTree (http://www.jstree.com/) がある．またブロックスタイルの操作性を意識して，木構造に基づく UI のパーツである部分木を格納するパーツパレットを設置する．そうして作成された木構造で表された正規表現から実コードを生成し，プログラミングエディタに挿入する．したがって本システムは，パーツパレット，木構造に基づく UI，プログラミングエディタから構成される．

ここで，エディタはブロックスタイルではない一般的なものとする．ブロックスタイルは用いず，正規表現を作成する際に必要であればツリー表示に基づく UI を活用するといった使い方を想定している．

3.　システム構成

本研究システムは Ajax ベースである．これは HTML5 や JavaScript のライブラリ，既存のオープンソースなシステムなどを利用し，要件を満足するシステムを効率良く開発するためである．また利用者にとっても，インストールやバージョンアップといった煩わしい作業を省けるなどの利点がある．

パーツを格納するパーツパレット，既存のプログラミングエディタ，木構造に基づく UI は，それぞれ jQuery UI，CodeMirror (http://codemirror.net/)，jsTree により実装した．jQuery UI とは jQuery (http://jquery.com/) が提供する UI で，jQuery とは JavaScript の標準的なユーティリティライブラリである．さきほど紹介した jsTree も jQuery を用いている．CodeMirror は既存の Web ベースエディタである．それら jsTree と CodeMirror 間に実装した正規表現の変換器を配置し，木構造に基づく UI で正規表現を作成する機能を開発した．図 3 が本システムのスクリーンショットである．

4.　システム評価

本システムと先行システムのそれぞれで十個の適当な正規表現を作成し，その際に要した操作（作成，

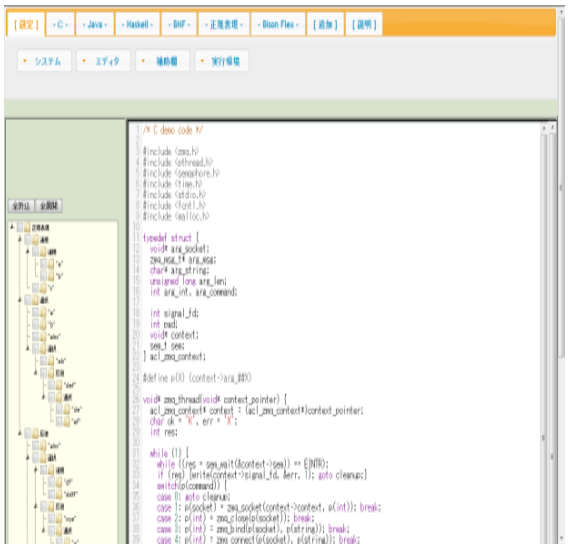

図 3　スクリーンショット

入力，移動，削除）の回数を調査した．その結果，本システムによる操作回数は先行システムの約 40%に抑えられていることが明らかになった．同様に，正規表現の表示に要したスペースの面積も調査したところ，本システムは先行システムの約 30%に抑えられているという結果になった．

5.　おわりに

今回はエクスプローラ風な UI でシステムを開発した．その結果，先行システムと比較し，ほどよい操作と表示を提供できた．

それでも，テキストで記述された正規表現と比較すると，表示に要した面積は大きい．このため，正規表現が長く，かつ，個数が増えた場合には，スペースを多く取り過ぎてしまう．そこで将来の課題として，ツリー表示を必要に応じて呼び出す仕組みを構築することが挙げられる．

現在，多層薄皮 UI “Tangerine” を構想している．これは，外見は通常のテキストエディタだが，内部的には構文の木構造を保持し，ドラッグ＆ドロップなどの特定の操作をしたときだけ木構造やブロック構造などの適切な UI を表示するといった支援を提供する．たとえば，正規表現のコード箇所にマウスオーバーした際に，ツリー表示を呼び出し，そこで正規表現を編集するなどである．

参考文献

(1) Maloney 他 :, “Scratch: A Sneak Preview,” Second International Conference on Creating, Connecting, and Collaborating through Computing. Kyoto, Japan, pp.104-109, 2004.
(2) 尾崎・香川：“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”, JSiSE 2013 講演論文集 TF1-1
(3) 白岩・香川：“Haskell に対する Web ベースのグラフィカルな学習支援環境” JSiSE2011 講演論文集, pp. 258—259