

1. 背景と目的

近年、関数型プログラミング言語が注目されている。その要因として関数型言語は並列処理に適しており、クラウド技術等の普及により近年必要性が増している高速な分散処理に適していることがある。しかし、関数型プログラミング言語は、基本的な文法が C などの手続き型の言語とは異なり、一般のプログラマには理解しづらいため、利用者は多くない。

本研究では一般のプログラマが理解しづらい関数型プログラミング言語の構造を視覚的に表示すればユーザーにとって理解しやすくなると考え、関数型プログラミング言語に基づくビジュアル言語の製作を目指した。

2. ビジュアルプログラミング言語

ビジュアルプログラミング言語とは、ソースプログラムをテキストで記述するのではなく、視覚的なオブジェクトで記述するプログラミング言語である。プログラマは、空間上でテキストやグラフィックシンボルを設置することでプログラムを記述していく。

多くのビジュアルプログラミング言語では、ノード(データや関数)とそれらを接続するアロー(矢印)を配置してプログラムを構成する。

3. Clojure

本研究では関数型プログラミング言語の中でも Clojure [1]のビジュアル言語の開発を行った。Clojure は JVM (Java Virtual Machine) 上で動作する関数型プログラミング言語であり、JVM 上で動作をするため Java の豊富な資産を使用することができるのが利点である。

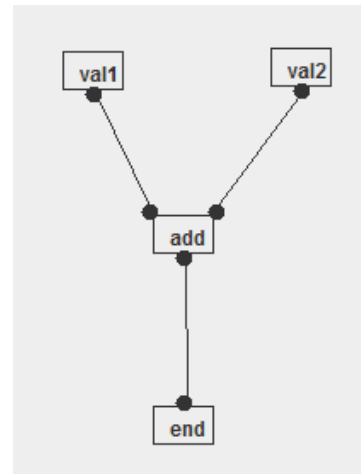


図1 ビジュアルエディタ

4. プログラムの構成

本研究で作成したプログラムは、ビジュアルエディタ、コードエディタ、ノードウィンドウの3つのウィンドウで構成されている。

4.1 ビジュアルエディタ

ビジュアルエディタの動作画面を図1に示す。ビジュアルエディタではビジュアルプログラミング言語の基本的な考えであるノードとそれらを接続するアローを配置してプログラムを構成する。

例えば図1のプログラムでは val1 と val2 という定数ノードと関数ノード add, そして end ノードが配置されている。val1 には値 3 が設定されており、val2 には値 7 が設定されている。ノード add には引数が 2 つ記述されており、関数ノードから end ノードに線を引くと定数や引数の関係がコードエディタに表示される。

ノードはマウスでドラッグすることにより、ビジュアルエディタ内であれば自由に移動さ

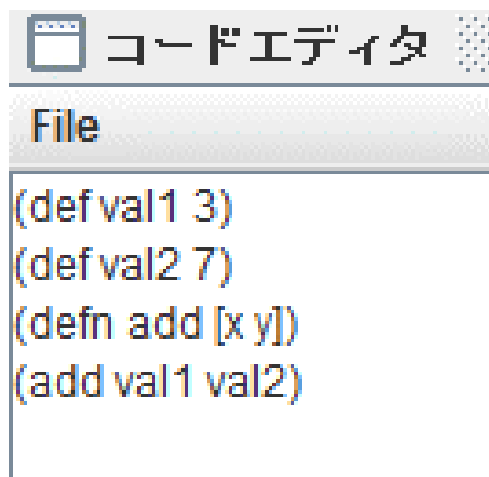


図2 コードエディタ

せることができる。

4.2 コードエディタ

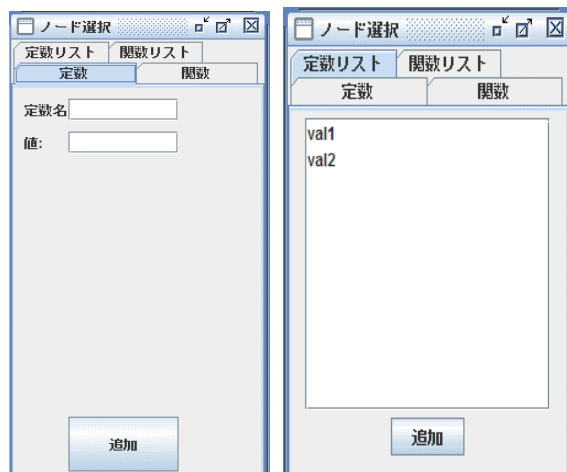
ビジュアルエディタで定数ノードや関数ノードを追加すると、同時に図2のコードエディタに定数名や、関数名などが表示される。本研究で作成したプログラムでは、関数内部のプログラムはこのコードエディタで記述・編集していく。コードエディタでは、テキストファイルの保存と読み込みができる。

4.3 ノードウィンドウ

ノードウィンドウを図3に示す。図3(a)は定数を入力するためのタブであり、ここに定数名と値を入力し、追加ボタンをクリックすると、ビジュアルエディタには定数ノード、コードエディタには Clojure のソースコードが追加される。

関数タブでは、関数名、引数1、引数2、をそれぞれ入力し、追加ボタンをクリックすることにより関数ノードと Clojure のソースコードがそれぞれのウィンドウに追加される。現在は引数が2つの関数しか対応していないため、引数の入力箇所は2か所となっている。

図3(b)は定数リストであり、定数タブで入力された定数が一覧表示される。定数リスト上で



(a)定数タブ

(b)定数リスト

図3 ノード選択ウィンドウ

定数名をクリックし、追加ボタンをクリックすることにより、同じ定数名のノードをビジュアルエディタに表示することができる。この例では、定数リストには val1 と val2 が表示されている。

5. まとめ

近年、プログラミング言語として関数型言語が注目されてきているが、一般のプログラマには関数型言語は理解しづらく利用者は多くない。そこで本研究では、関数型言語 Clojure の考えに基づいたビジュアル言語を製作し、定数や関数の可視化を行った。

本研究の課題として、現在は関数が2引数のものしか対応できていないので、1引数や3引数以上のものにも対応する必要がある。

関数型言語では繰り返し文を用いずに再帰を用いてコードを記述することが多いが、再帰関数をどのようにビジュアルエディタ上で表現するかも課題である。

参考文献

[1] S. Halloway and A. Bedra: プログラミング Clojure 第2版, オーム社, 2015.