

意思決定のための
ビジュアルプログラミングによる
データ分析を支援する
Web アプリケーションの開発

Developing Web Applications to Support Data Analysis
by Visual Programming for Decision Making

Keniti Numata
u055017@st.pu-toyama.ac.jp

Graduate School of Engineering
Department of Information Systems Engineering
Toyama Prefectural University

Teams, 9:50-10:15 Friday., December 4, 2020.

1. はじめに
2. デジタルトランスフォーメーション
3. ビジュアルプログラミング言語
4. 修論に向けて

1.1 本研究の背景

2/14

- 1. はじめに
- 2. デジタルトランスフォーメーション
- 3. ビジュアルプログラミング言語
- 4. 修論に向けて

背景

近年、企業などでは Society5.0 に向けた世間に溢れる様々な情報を収集し、ビッグデータとして様々な処理や分析によって情報を扱うことが増えている。これらのデータは、膨大であるため一般的にプログラミング言語を使って処理される。

ビッグデータについて

- 1 ビッグデータとは、大量で高頻度な多様性があるデータとして定義される。
- 2 ビッグデータの例として、ライフログデータや顧客データ、購買データなどがある。
- 3 需要の予測やコストの削減、仕入れなどの最適化など様々な活用方法がある。

1.2 本研究の目的

3/14

1. はじめに
2. デジタルトランスフォーマーション
3. ビジュアルプログラミング言語
4. 修論に向けて

現在の問題点

- 1 ユーザビリティの低さ
プログラミング初心者にとって、プログラミングは手につけづらい。
- 2 データの取り扱い
データを簡単に分析する手法が存在しない。

本研究の目的

- ・プログラミング初心者でも扱いやすい環境の開発
- ・データを処理できるように開発
- ・外部に公開し、複数人から利用できるようにする。

2.1 Society 5.0 と web サービス

4/14

1. はじめに
2. デジタルトランスフォーメーション
3. ビジュアルプログラミング言語
4. 修論に向けて

Society 5.0

Society 5.0 とは、サイバー空間（仮想空間）とフィジカル空間（現実空間）を高度に融合させたシステムにより、経済発展と社会的課題の解決を両立する人間中心の社会のことである。

Society 5.0 と web サービス

サイバー空間上で web サービスとして Blockly を提供し、ユーザーがフィジカル空間上でデータを分析できるようにすることで Society 5.0 を実現する。

また、このような IT を利用した人々の生活をあらゆる面でより良い方向に変化させるようなことをデジタルトランスフォーメーションという。

2.2 サーバサイドプログラミング

5/14

1. はじめに
2. デジタルトランスフォーメーション
3. ビジュアルプログラミング言語
4. 修論に向けて

サーバサイドプログラミング

本研究では、HTTP 通信を行ったあとにユーザーからの入力に対しての処理を、非同期通信を使って Web サーバ上でプログラムの実行が要求され、結果をウェブブラウザに対して送信するシステムを開発する。

クライアントサイドコードに使うことができる言語は、HTML,CSS,Javascript であるが、サーバーサイドによる処理を挟むことでそれ以外の言語 (Perl,PHP,Python,Ruby など) を使う事ができるようになる。

3.1 ビジュアルプログラミング言語

6/14

ビジュアルプログラミング言語

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のこと。視覚的でわかりやすいものが多いため、プログラムの組み立て方を学ぶのに有効であると注目されている。

1. はじめに
2. デジタルトランスフォーマー
3. ビジュアルプログラミング言語
4. 修論に向けて

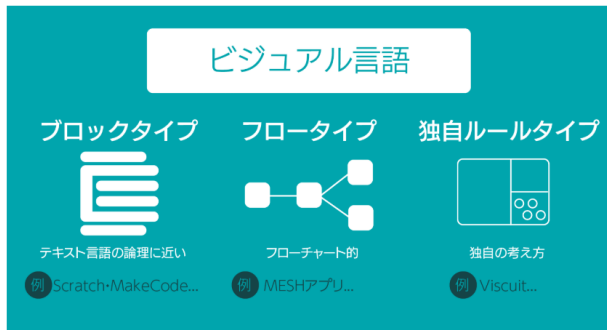


Figure 1: ビジュアルプログラミング

3.2 ブロックタイプの ビジュアルプログラミング言語

7/14

ブロックタイプのビジュアルプログラミング言語

機械学習（人工知能・AI）を使って課題を解決するクラウドサービスの MAGELLAN BLOCKS（BLOCKS）や教育用作られ様々なアプリケーションに応用して使われている Blockly などがある。応用して使われているサービスとして Scratch や MakeCode が存在する。

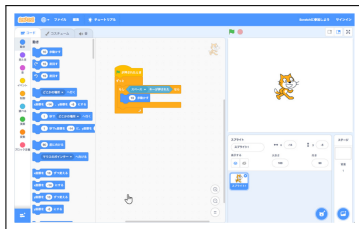


Figure 2: Scratch

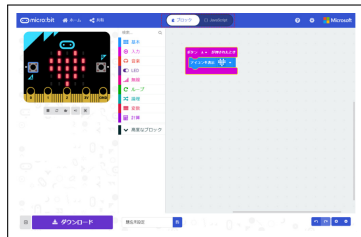


Figure 3: MakeCode

1. はじめに
2. デジタルトランスフォーマー
アプリケーション
3. ビジュアルプログラミング言語
4. 修論に向けて

3.3 ビジュアルプログラミング言語 (Blockly)

8/14

Blockly

Google が提供しているビジュアルプログラミング言語のライブラリ。簡単な記述で自分だけのビジュアルプログラミング言語を作ることができる。

また、カスタムブロックという機能があり、もともとあるブロックの他にユーザが好きなブロックを作成することができる。



Figure 4: Blockly

1. はじめに
2. デジタルトランスフォーメーション
3. ビジュアルプログラミング言語
4. 修論に向けて

3.4 カスタムブロック

9/14

3.4.1 ブロックの定義

作成したいブロックの外観とブロックに接続する数値やテキストをここで定義する。

外観は、ブロックの色やブロックの接続 (構文ブロックと値ブロック), 表示する文字等がある。また、ブロック内の空きに何を入力 (input) として何を出力 (output) とするかなど決める。

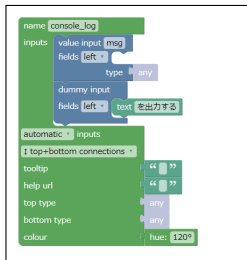


Figure 5: console log に結果を出力するブロックの定義

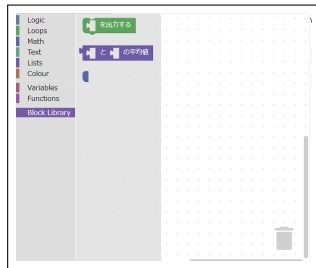


Figure 6: toolbox

1. はじめに
2. デジタルトランスフォーマーション
3. ビジュアルプログラミング言語
4. 修論に向けて

3.2 カスタムブロック

10/14

3.4.2 コードの生成

コードの生成では、ブロックの動作の定義を行う。例えば、平均値を出すブロックを作成するときは、平均を出す計算部分を動作の定義で書く。

```
Blockly.JavaScript['average'] = function(block) {  
  var value_v1 = Blockly.JavaScript.valueToCode(block, 'v1', Blockly.JavaScript.ORDER_ATOMIC);  
  var value_v2 = Blockly.JavaScript.valueToCode(block, 'v2', Blockly.JavaScript.ORDER_ATOMIC);  
  // TODO: Assemble JavaScript into code variable.  
  var code = '(' + value_v1 + '+' + value_v2 + ')/2';  
  // TODO: Change ORDER_NONE to the correct strength.  
  return [code, Blockly.JavaScript.ORDER_NONE];  
};
```

Figure 7: 動作の定義

3.4.3 ブロックのカテゴリーと配置決め

作ったブロックをどこのカテゴリーに入れるかを決める。

1. はじめに
2. デジタルトランスフォーマーション
3. ビジュアルプログラミング言語
4. 修論に向けて

今後の方針

1 blockly の分析と出力ブロックの修正

- ▷ 分析ブロックの設定値の指定
- ▷ 二次元散布図ブロックの修正
- ▷ 回帰分析ブロックの修正

2 修論に向けた新規性の了見

1. はじめに
2. デジタルトランスフォーマーション
3. ビジュアルプログラミング言語
4. 修論に向けて

1.blockly の分析と出力ブロックの修正

12/14

1.1 分析ブロックの設定値の指定

分析ブロックの設定値を変更することで恣意的なプログラムと見受けられないにする。

設定値の入力は、直接数値をいれるかプルダウンで選択する。

ex1) 周波数成分を表示するブロックでサンプリング周波数の値を変更できるようにしたりする。

ex2) デンドログラムのクラスター間の距離測定方法を複数用意する。

1. はじめに
2. デジタルトランスフォーマーション
3. ビジュアルプログラミング言語
4. 修論に向けて

1.blockly の分析と出力ブロックの修正

13/14

1.2 二次元散布図ブロックの修正

エラーを特定して、軸を指定して二次元散布図を表示できるように修正する。

1.3 回帰分析ブロックの修正

単回帰分析しかできないブロックから、重回帰分析もできるようなブロックにする。

使う変数は、直接入力またはプルダウンで設定できるようにする。

1. はじめに
2. デジタルトランスフォーメーション
3. ビジュアルプログラミング言語
4. 修論に向けて

2. 修論に向けた新規性の了見

14/14

修論に向けた新規性の了見

- ▷ データ処理速度の改善.
- ▷ データ解析ブロックを増やす. 動画が扱えるブロックとか.
- ▷ ブロックを縦積みで連結し, より複雑な解析もできるようにする.

1. はじめに
2. デジタルトランスフォーメーション
3. ビジュアルプログラミング言語
4. 修論に向けて