

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. 進捗具合
4. おわりに

# 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](mailto:u055017@st.pu-toyama.ac.jp)

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

# 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. 進捗具合
4. おわりに

## 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、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  
スフォーメー  
ション
3. ビジュアルプロ  
グラミング言語
4. 進捗具合
4. おわりに

## 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かけづらい。

### 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. 進捗具合
4. おわりに

### 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. 進捗具合
4. おわりに

### 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

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. 進捗具合
4. おわりに

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。

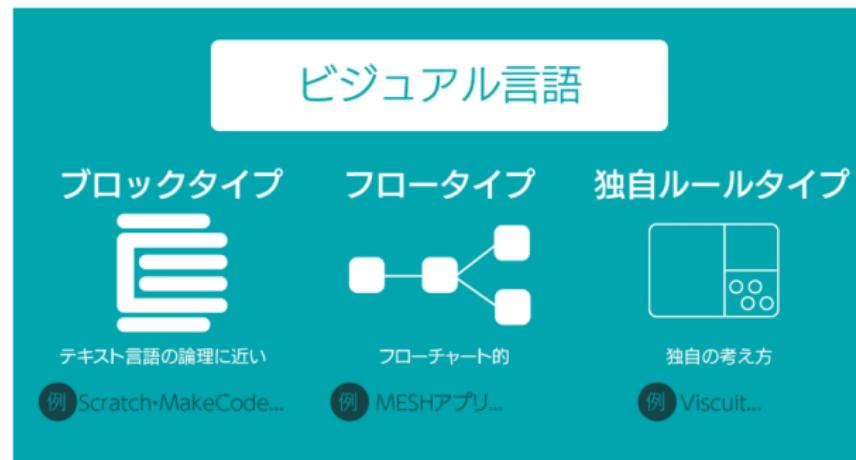


Figure 1: ビジュアルプログラミング

## 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。

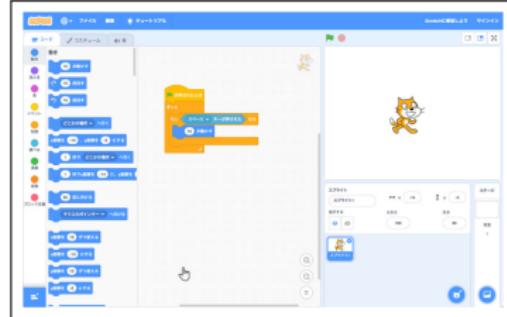


Figure 2: Scratch



Figure 3: MakeCode

1. はじめに
2. デジタルトランジスタフォーメーション
3. ビジュアルプログラミング言語
4. 進捗具合
4. おわりに

# 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

- はじめに
- デジタルトラン  
スフォーメー  
ション
- ビジュアルプロ  
グラミング言語
- 進捗具合
- おわりに

# 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。

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. 進捗具合
4. おわりに

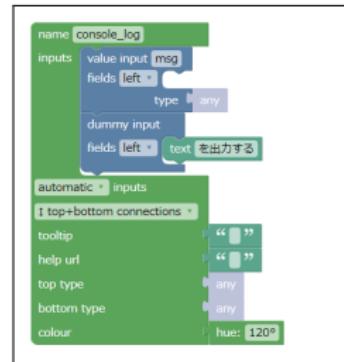


Figure 5: console log に結果を  
出力するブロックの定義

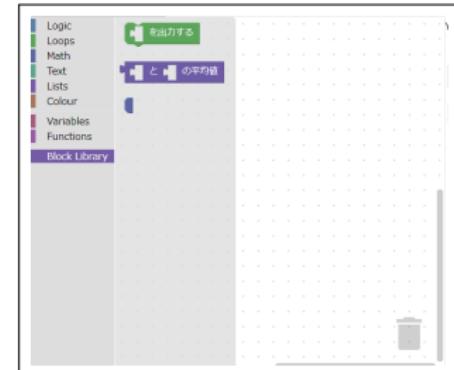


Figure 6: toolbox

## 3.2 カスタムブロック

10/14

1. はじめに
2. デジタルトランシフーメーション
3. ビジュアルプログラミング言語
4. 進捗具合
4. おわりに

### 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。

```
average.js > ...
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。

# 進捗

11/14

## ゼミで作成してもらったブロックをあわせる

先に完成でできた決定木分析と主成分分析とナイーブベイズと線形計画問題の計算するブロックを自分の環境に統合させることにした。

ナイーブベイズ以外のブロックは実装して動くことを確認できた。ナイーブベイズは、必要なモジュールである MeCab の設定を間違えた。sudo が使えなくなった。



Figure 8: 統合したブロック

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. 進捗具合
4. おわりに

## CSV の読み込みブロックの改良

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. 進捗具合
4. おわりに

これまでの CSV は UTF-8 のみで読み込みを想定していたが、分析手法が増え日本語を含む CSV を読み込むと文字化けが起きた。最初にファイルを UTF-8 のテキストとして読み込むプログラムから、最初にファイルをバイナリで読み込んで UTF-8 に変換するプログラムに変更した。

バイナリ配列から文字に変換するのに `encoding.min.js` を使用した。

## CSV の出力ブロックの改良

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. 進捗具合
4. おわりに

CSV 読み込みブロックと同様に、出力に関しても日本語を含むことがあります。現状文字化けして出力されてしまうのでバイトオーダーマーク (BOM) を設定することで文字化けしないようにしました。

BOM とは、Unicode の符号化形式で符号化したテキストの先頭につける数バイトのデータのことである。

このデータを元に Unicode で符号化されていることおよび符号化の種類の判別に使用する。

# おわりに

14/14

## 今回の進捗

- ① 決定木分析と主成分分析とナイーブベイズと線形計画問題の計算するブロックを追加した。
- ② CSV ファイルの入出力ブロックを改良した。

## 今後の課題

- ① ツールボックスのカテゴリを増やし、ブロックの色を変える
- ② プルダウンで分析手法を変えるブロックを完成させる。
- ③ ブロックの説明が必要なものにブロックの解説を乗せるようにする。