

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

Graduate School of Information Engineering, Toyama Prefectural University  
u055017@st.pu-toyama.ac.jp

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

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. 提案手法とシステムのアーキテクチャ
5. 実験結果ならびに考察
5. おわりに

# 1.1 本研究の背景

2/21

## 背景

近年、企業などでは世間に溢れる様々な情報を収集し、ビック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。

## ビックデータについて

- 1** ビックデータとは、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** ビックデータの例として、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. 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. 提案手法とシステムのアーキテクチャ

5. 実験結果ならびに考察

5. おわりに

## 1.2 本研究の目的

3/21

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. 提案手法とシステムのアーキテクチャ

5. 実験結果ならびに考察

5. 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

4/21

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. 提案手法とシステムのアーキテクチャ
5. 実験結果ならびに考察
5. 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

5/21

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. 提案手法とシステムのアーキテクチャ
5. 実験結果ならびに考察
5. 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

6/21

### 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。

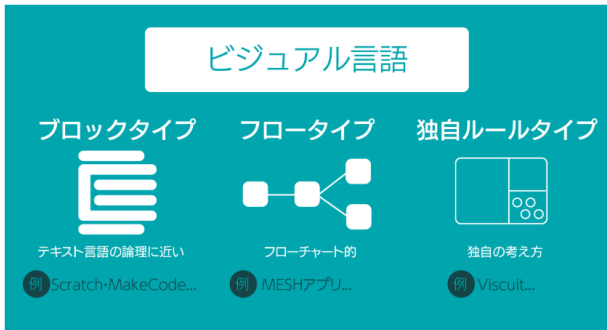


Figure 1: ビジュアルプログラミング

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. 提案手法とシステムのアーキテクチャ
5. 実験結果ならびに考察
5. おわりに

## 3.1 ビジュアルプログラミング言語

7/21

### 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。

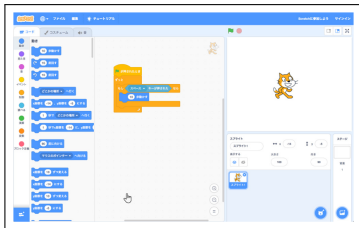


Figure 2: scratch



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. 提案手法とシステムのアーキテクチャ
5. 実験結果ならびに考察
5. おわりに

## 3.1 ビジュアルプログラミング言語

8/21

### 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、作成したブロックは JavaScript や python, PHP, Lua, Dart などのプログラミング言語にエクスポートすることができる。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。

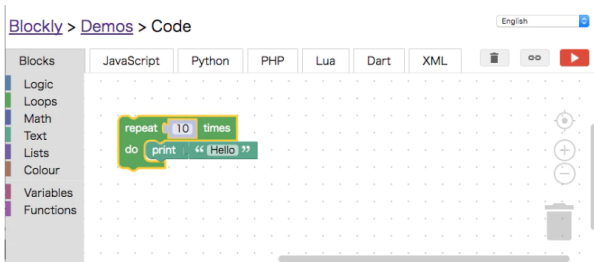


Figure 4: Blockly



## 3.2 カスタムブロック

9/21

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. 提案手法とシステムのアーキテクチャ
5. 実験結果ならびに考察
5. おわりに

### カスタムブロックの構成

1. ブロックの定義
2. コードの生成
3. ブロックのカテゴリーと配置決め

### Blockly Developer Tools

簡単にカスタムブロックを作成する支援ツールとして、Blockly Developer Tools がある。  
この支援ツールは blockly を用いて、ブロックを作ることができる。

## 3.2 カスタムブロック

10/21

### 3.2.1 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)or 出力 (output) とするか決める。



Figure 5: block の種類



Figure 6: console log に結果を出力する関数

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. 提案手法とシステムのアーキテクチャ
5. 実験結果ならびに考察
5. おわりに

## 3.2 カスタムブロック

11/21

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. 提案手法とシステムのアーキテクチャ
5. 実験結果ならびに考察
5. おわりに

### 3.2.2 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

## 3.2 カスタムブロック

12/21

### 3.2.3 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定める。

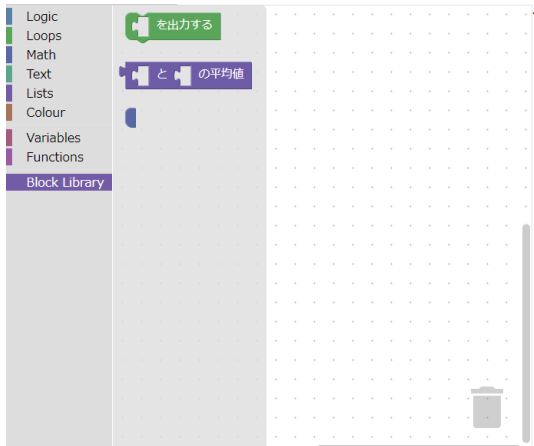


Figure 8: toolbox

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. 提案手法とシステムのアーキテクチャ
5. 実験結果ならびに考察
5. おわりに

## 4.1 提案手法

13/21

### 従来の Blockly の問題点

- データを分析するとき、外部からデータを読み込む必要があるが blockly のデモコードには外部からのファイルの読み込みが実装されていない点。
- blockly はブラウザをベースにしてクライアント側で javascript によって処理を行うサーバ・サイド・インクルード (SSI) であるため、データ分析関連の処理をできるライブラリを使うことができない点。

### 提案手法

- CSV ファイルの読み込みブロックの作成
- Ajax による通信
- 機械学習ライブラリの導入

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. 提案手法とシステムのアーキテクチャ
5. 実験結果ならびに考察
5. おわりに

## 4.2 Ajax による非同期通信

14/21

### Ajax 通信とは

通常, HTTP とは Web ページなどをやり取りする技術 (プロトコル) で, ブラウザとサーバーとの間で使われている. Ajax を使うと, ブラウザに代わって JavaScript がサーバーと HTTP 通信することができる.

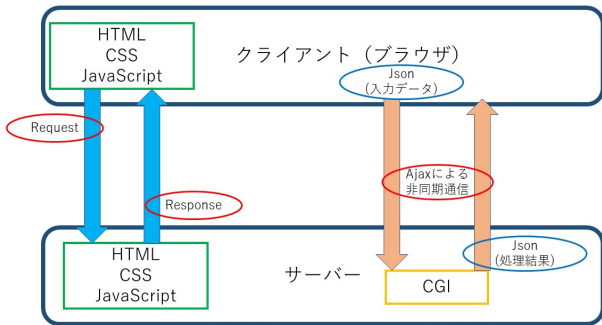


Figure 9: 処理の遷移

## 4.3 CSV ファイルの読み込みブロックの作成

15/21

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. 提案手法とシステムのアーキテクチャ
5. 実験結果ならびに考察
5. おわりに

### 入力データの形式

入力データは，データの入力形式をあらかじめ統一し，拡張子は CSV とすることにした．

入力形式としては，以下のようにした．

- セルの 1 行 1 列目に入力データ数
- 2 行 1 列目に説明変数，2 行 2 列目に目的変数
- 3 列目にラベル名
- 4 列目に基本データ型
- 5 列目以降に 3,4 列目に対応する入力データ

## 4.4 機械学習ライブラリの導入

16/21

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. 提案手法とシステムのアーキテクチャ
5. 実験結果ならびに考察
5. おわりに

### CGI

Ajax による通信から先のバックエンドでは、CommonGateway Interface(CGI) と呼ばれるプログラムを使って、クライアントから送られたデータを処理して、結果をクライアントに返す。CGI は python で記述し、scikit-learn と呼ばれる機械学習ライブラリを本研究では使用した。



## 5.1 作成したブロック

17/21

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. 提案手法とシステムのアーキテクチャ
5. 実験結果ならびに考察
5. おわりに

### ファイルを読み込むブロック

ファイルを読み込むブロックの見た目は以下の図である．\*\*ファイルを選択してください\*\*の枠内をマウスでクリックすることで、ポップアップでエクスプローラーが開き、そこから入力データとなる csv ファイルを選択すると、入力データを json に置き換えてこのブロック自体に内容を保持させることができる．

**\*\* ファイルを選択してください \*\*** を読み込み

Figure 10: ファイルを読み込むブロック

## 5.1 作成したブロック

18/21

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. 提案手法とシステムのアーキテクチャ
5. 実験結果ならびに考察
5. おわりに

### 線形回帰のブロック

データ分析は線形回帰の計算を対象とするブロックを以下の図のように作成した．このブロックの使い方としては，ブロックの空いたところに入力データとなる json が含まれるブロックを入れることで実行時に線形回帰の計算を行なわれ，回帰係数，切片，決定係数といった結果が帰ってくるものとなっている．



Figure 11: 線形回帰のブロック

## 5.1 作成したブロック

19/21

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. 提案手法とシステムのアーキテクチャ
5. 実験結果ならびに考察
5. おわりに

### 出力結果のダウンロードができるブロック

出力結果のダウンロードができるブロックを以下の図のように作成した．このブロックの使い方としては，ブロックの空いたところを入力として，プログラムの実行時に自動で入力部分の処理結果を csv に変換し，そのファイルのダウンロードが開始されるものとなっている



Figure 12: 出力結果のダウンロードができるブロック

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. 提案手法とシステムのアーキテクチャ
5. 実験結果ならびに考察
5. おわりに

### 考察

今回作成したブロックを以下の図のように組み合わせることで、blockly で入力データに対して線形回帰の計算をし、結果を CSV ファイルでダウンロードできるようになった。



Figure 13: 作成したブロックを組み合わせた図

## 5. おわりに

21/21

### まとめ

本研究ではビジュアルプログラミング言語によるデータ解析システムの開発をした。Ajax 通信を使って blockly 上で線形回帰の計算ができるようになった。

### 今後の課題

- データを分析する手法を調べ，そのブロックを作成していく。
- 今回作った線形回帰のブロックを編集してオプションで単回帰分析か重回帰分析をブロックで選択できるようにする。

1. はじめに
2. デジタルトランスフォーマーション
3. ビジュアルプログラミング言語
4. 提案手法とシステムのアーキテクチャ
5. 実験結果ならびに考察
5. おわりに