

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. 進捗状況
4. おわりに

# 1.1 本研究の背景

2/17

## 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. 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. 進捗状況

4. おわりに

## 1.2 本研究の目的

3/17

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. 進捗状況
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

4/17

- 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. 進捗状況
- 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

5/17

- 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. 進捗状況
- 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

6/17

### 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。

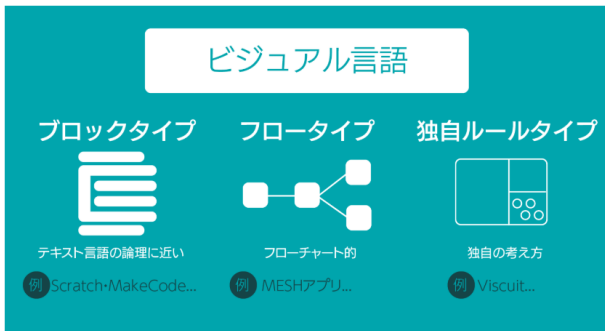


Figure 1: ビジュアルプログラミング

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. 進捗状況
4. おわりに

## 3.2 ブロックタイプの ビジュアルプログラミング言語

7/17

### 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。

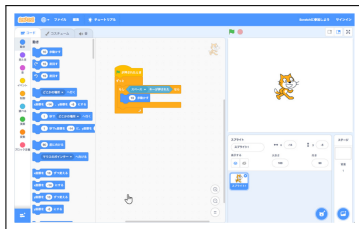


Figure 2: Scratch

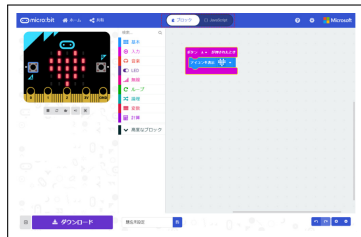


Figure 3: MakeCode

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. 進捗状況
4. おわりに

### 3.3 ビジュアルプログラミング言語 (Blockly)

8/17

#### 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。

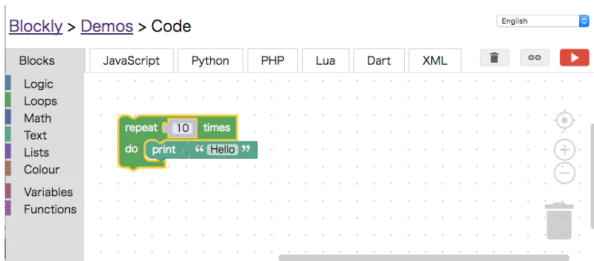


Figure 4: Blockly

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. 進捗状況
4. おわりに



## 3.4 カスタムブロック

9/17

### 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。

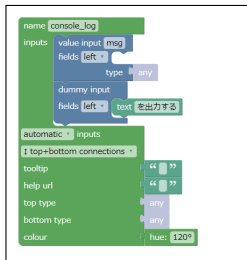


Figure 5: console log に結果を出力するブロックの定義

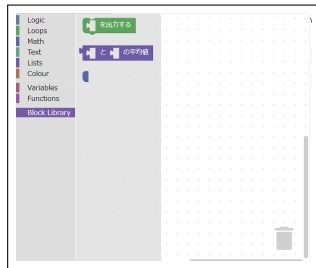


Figure 6: toolbox

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. 進捗状況
4. おわりに

## 3.2 カスタムブロック

10/17

### 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. 進捗状況
4. おわりに

## 今回の進捗

- 1 ブロックの解説
- 2 Blockly のデザインの編集
- 3 ワークスペース読み込みの修正
- 4 ログイン後のサイトの作成

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. 進捗状況
4. おわりに

# 1. ブロックの解説

12/17

## 1.1 ブロックの解説

解説を知りたいブロックに対してカーソルを当てることで、解説が出るようにした。 tooltip を編集した。

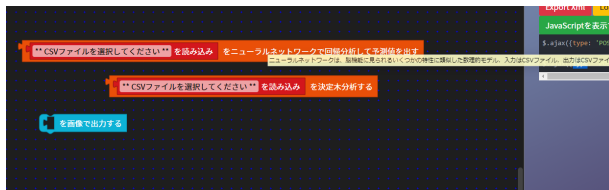


Figure 8: ブロック解説

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. 進捗状況
4. おわりに

## 2.Blockly のデザインの編集

13/17

### 1.2 Blockly のデザインの編集

- ① フォントを変更した。使ったフォントは、Noto Sans JP。これは、Google が Adobe と共同開発したフォントファミリー。
- ② 画面が白基調で使っていると目が疲れるので黒基調に css を変更した。workspace の背景色は、1e1e1e。html と toolbox の背景色は、2e0736 から 94c3e2 にグラデーションしてカッコよくした。
- ③ toolbox のセパレータで区切られたカスタムブロックの項目に、Custom Blocks というタイトルをつけて作成したブロック達をわかりやすくした。
- ④ toolbox のボタンサイズを大きくし、ブロックの画像を入れることで視覚的に分かりやすく、また選択ボタンを押しやすくした。

## 2.Blockly のデザインの編集

14/17

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. 進捗状況
4. おわりに

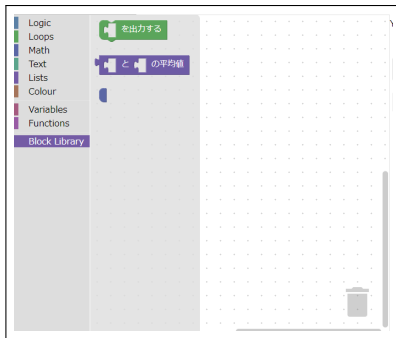


Figure 9: 古いツールボックス



Figure 10: 新しいツールボックス

### 3. ワークスペース読み込みの修正

15/17

#### 3. ワークスペース読み込みの修正

ワークスペース上のブロックの配置を保存できるようにした．エラーも修正した．出力は，xml 形式で中身にブロックの種類と座標が保存されている．



Figure 11: workspace

- 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. 進捗状況
- 4. おわりに

## 4. ログイン後のサイトの作成

16/17

### ログイン後のサイトの作成

ログイン後のサイトの作成を横井さんが作ってくれました。



Figure 12: ログイン前のサイト

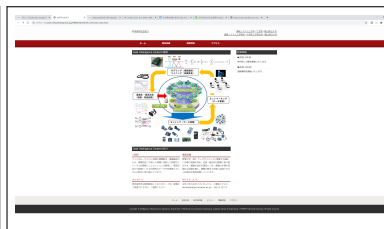


Figure 13: ログイン後のサイト

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. 進捗状況
4. おわりに



## 今回の進捗

- ① ブロックの解説
- ② Blockly のデザインの編集
- ③ ワークスペース読み込みの修正
- ④ ログイン後のサイトの作成

## 今後の課題

- 1 エラーが起きないか検証する.