

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

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

1.1 本研究の背景

2/22

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

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

3/22

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

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

4/22

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

5/22

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

6/22

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. 進捗状況
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。

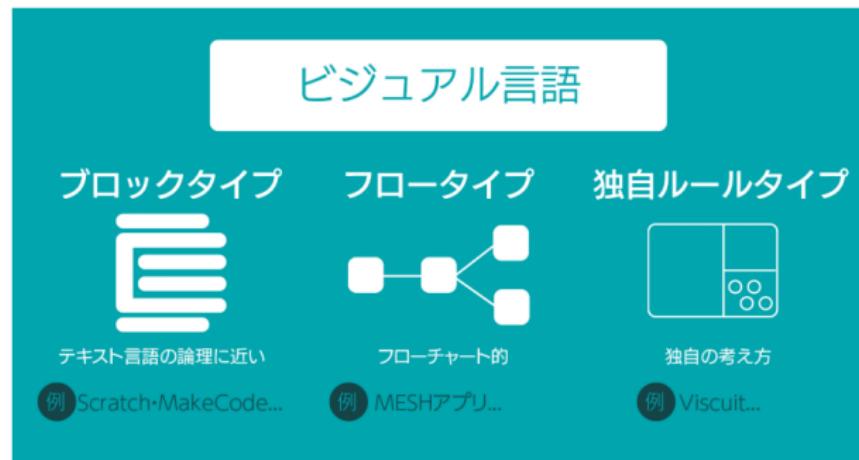


Figure 1: ビジュアルプログラミング

3.2 ブロックタイプのビジュアルプログラミング言語

7/22

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。

1. はじめに
2. デジタルトランシ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

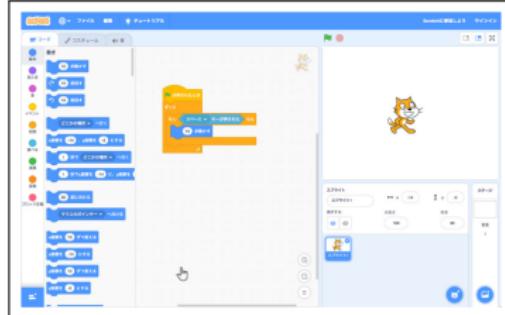


Figure 2: Scratch



Figure 3: MakeCode

3.3 ビジュアルプログラミング言語 (Blockly)

8/22

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

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

3.4 カスタムブロック

9/22

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。

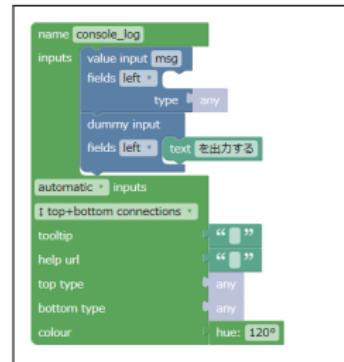


Figure 5: console log に結果を出力するブロックの定義

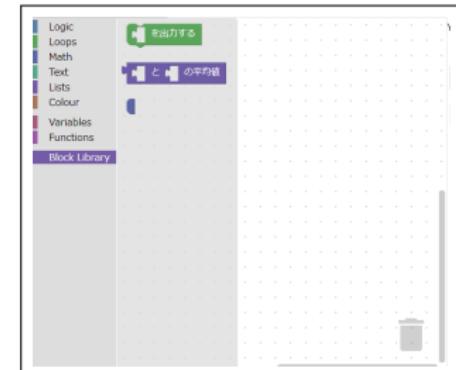


Figure 6: toolbox

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

3.2 カスタムブロック

10/22

1. はじめに
2. デジタルトランシ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

3.4.2 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。

進捗報告一覧

11/22

今回の進捗

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** ゼミのブロックの確認と編集
- 2** Blockly の見た目の編集
- 3** 学外からアクセス

1. ゼミのブロックの確認と編集

12/22

1.1 ゼミのブロックの確認状況

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 ナイーブベイズ

前回失敗した MeCab のインストールが成功し、動くようになった。

② 文章の類似度

うまく動いたが出力結果の表示が少しおかしくなっていた。

③ 画像の類似度

動いた。

④ 文字認識

動いた。

1. ゼミのブロックの確認と編集

13/22

1.2 ゼミのブロックの編集

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

画像を扱うブロックが出てきたので、画像を読み込むブロックの作成を行った。画像の類似度を求めるブロックが、target 画像に対して複数の画像の類似度を出せるように画像を複数読み込めるようにした。

ゼミのブロックは、手動で base64 型にしたもの CSV に貼り付けたものが入力だったので、今回作成したブロックと対応するように編集した。

画像の類似度ブロックは、一枚の画像としか比較できなかつたので複数の画像と比較できるように編集した。

また、文字認識のブロックは tesseract layout の値を 6 に変更して文字認識の精度を上げた。文字認識の結果はテキストで表示したくなつたので、テキストで出力するブロックを作った。

** CSVファイルを選択してください ** を読み込み

 をテキストで出力する

Figure 8: 新しいブロック

2. Blockly の見た目の編集

14/22

Blockly の見た目の編集

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 入力プレビューが長かったので、ブロックの文字表示数を固定した。



Figure 9: 固定長ブロック

- ② 回帰分析のブロックを一つにまとめた。



Figure 10: 新しい回帰分析ブロック

2. Blockly の見た目の編集

15/22

Blockly の見た目の編集

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 FetchAPI によって、ページ下部に取得したツールボックスの文字が表示していたのを直した



Figure 11: 邪魔なやつ

2. Blockly の見た目の編集

16/22

Blockly の見た目の編集

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

- ④ bootstrap のグリッドを固定幅コンテナから可変幅コンテナに
変更した

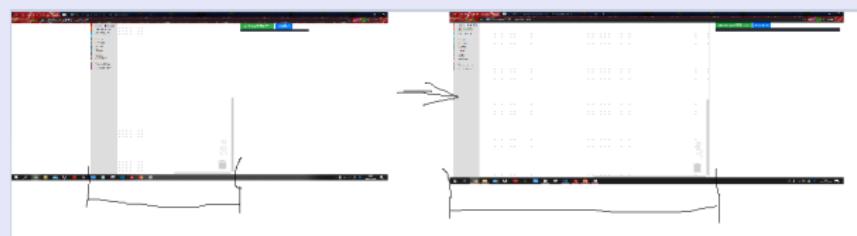


Figure 12: グリッドの種類の変更

2. Blockly の見た目の編集

17/22

Blockly の見た目の編集

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

- ⑤ blockly のカテゴリを入力と分析と出力の 3 種類に分けた
- ⑥ カテゴリ別にブロックの色を変えた
- ⑦ 先に入力ブロックをはめ込んだ分析系のブロックをツールボッ
クスに入れた。
- ⑧ ブロックの彩度と明度を変えた。

2. Blockly の見た目の編集

18/22

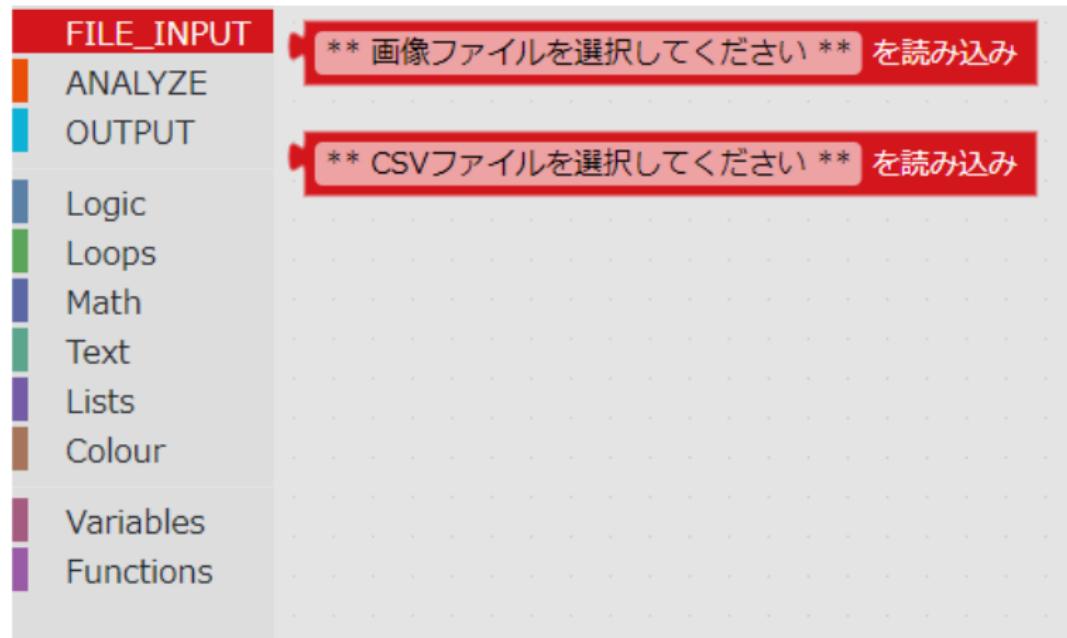


Figure 13: 入力ブロック

2. Blockly の見た目の編集

19/22

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



Figure 14: 分析ブロック

2. Blockly の見た目の編集

20/22

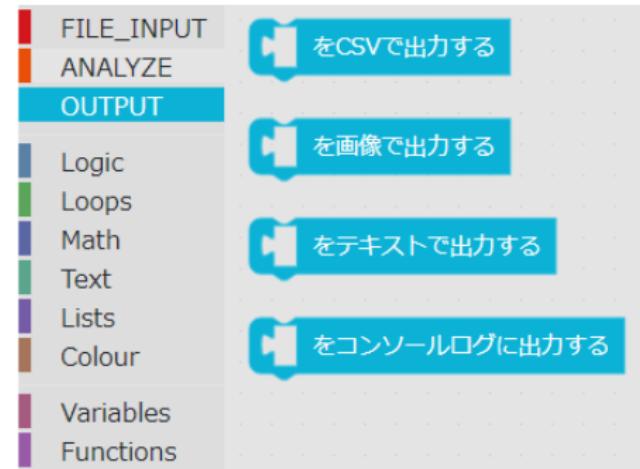


Figure 15: 出力ブロック

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. 進捗状況
4. おわりに

3. 学外以外

21/22

研究室サーバーに学外からアクセスできるようになった。

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

おわりに

22/22

今回の進捗

- ① ゼミのブロックの確認と編集
- ② Blockly の見た目の編集
- ③ 学外からアクセス

今後の課題

- ① 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。