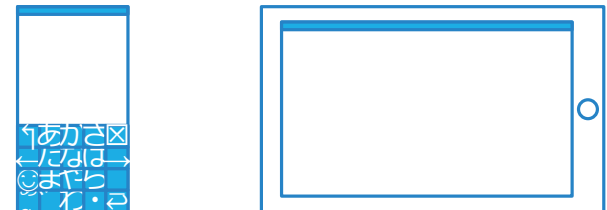


タブレットでの プログラミングを支援する 雛形作成システム

学籍番号：1421083 氏名：栗原 準
指導教員：鷹野孝典

研究背景

- プログラミングをする上で、PC環境は基本的に必須であり、エディタやIDEを使用する。
- スマートフォンやタブレットなど、画面が小さく手軽な端末でもある程度プログラミングができる。
- しかし、これらの端末においてのソフトウェアキーボードでは、物理キーボードに比べると十分なタイピングが行えない。
 - しかし、タッチ操作による直感的な操作が行える。
- プログラミングの設計はどのようにロジックを組み立てていくかなど、設計する人の負担が大きい



関連研究(1)

スニペット推薦

1. 回帰結合ニューラルネットワークを利用したAPI推薦手法(情報処理学会論文誌 2017)
 - 回帰結合ニューラルネットワーク(RNN)を利用したAPI推薦, メソッドの呼び出し順序に着目
2. 「関心度に基づいたソースコード推薦システム」(研究報告ソフトウェア工学2014)
 - 類似コード断片をランク付けして推薦



関連研究(2)

タブレット活用

1. 「プログラミング入門教育におけるペンタブレットの効果とモチベーションの関係」 (第75回全国大会講演論文集 2013)
2. 「タブレット端末を活用したプログラミング教育」 (名古屋文理大学紀要 2013)
3. 「タブレット端末で動作する, インタプリタ型言語搭載マイコンのプログラミング環境の開発」 (研究報告コンピュータと教育 2013)

タブレットを活用した研究はいくつも存在しており, タブレットでの開発はハードルが低くなったり, 利便性が高いなどのメリットがある。



研究動機

- 設計が重要なプログラミングにおいて、タブレット等の手軽な携帯端末から雛形の作成を行うことで、場所を問わず設計が可能になる。
 - 電車やバスの中だと、わずかな空き時間でPCを出すことは難しくても、タブレットを出す余裕くらいならある

研究課題

- スマートフォンやタブレットでは構文の入力が困難な場合があり、PC上のプログラミングと同等のことをしようとする操作性に限界がある。
- プログラミング開発時においていくつかの手法が存在した時、それらをよく吟味してから比較する必要がある、時間がかかる。
 - 例えばJavaScriptでの非同期通信での情報取得は方法が複数ある。

研究目的

- タブレットで場所を選ばずにソースコードの雛形の作成を行い、PC環境下での設計時の負担を軽減するようなエディタ及びシステムを提案する。
- プログラミングにおいて「ある課題を成し遂げたい」ときに幾つかの手法があったときに、それぞれを吟味する必要があり、その検索推薦をサポート



- 雛形: 寄せ集めたソースコードの断片の集合体。
実際のソースコードを書くときにテンプレートとなりうるもの

アプローチ - UI

1. 雛形作成時, ソースコードの断片 (スニペット) を検索する.
 - 検索はクロールされたデータを元にし, エディタ上に挿入をすることで, プログラミングの効率化を図る.
2. 検索したスニペットをブロック (後述) としてファイルに挿入する.
3. 生成されたソースコードの雛形を, 実際のプログラミングの資源として活用できる.

ブロックの詳細

- ブロックという概念を導入する
 - スニペットをブロックという単位で定義する。ブロックは並び替えや追加・削除ができる。
- タッチ操作による直感的なブロックの並び替えが可能。

UIの詳細



ファイルリスト

- ファイルの選択ができる

検索画面

- キーワード検索して結果を左側のブロッケー覧にドラッグ&ドロップすることでソースコードを挿入できる

ブロッケー覧と追加ボタン

- 選択したファイル进行处理ごとに分けてブロックで表現
- クリックするとそのブロックのソースコード編集画面に移動する
- ブロックは並べ替えが可能
- ブロックの追加ボタンで空のブロックを追加可能。タイトルから検索や推薦を行う

アプローチ – 検索

- Qiita, Stack Overflow, Githubと言ったソースコードが掲載されているサイトをクロールして, スニペットを抽出する.
 - 多くの場合はコードとともにそれが何をするかと言ったメタ情報 (ソースコードのコメントや記事中の解説) が含まれている.



^{†1}:スライド3の1 ^{†2}:UCI Source Code Data Sets, 多くのリポジトリデータの集合

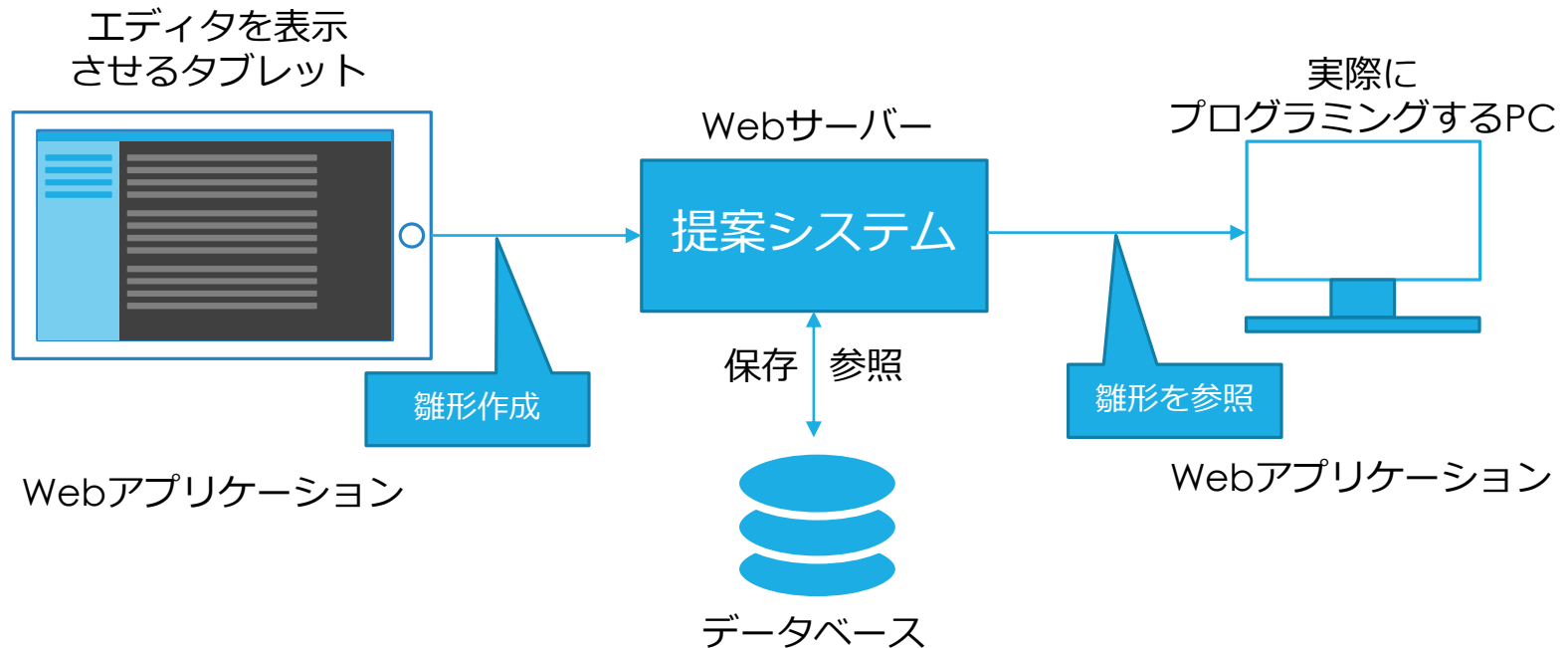
アプローチ – 検索(2)

- 検索のクエリには単純なテキストによる検索だけではなく、ブロックの前後のつながりをコンテキストとして利用する
 - 連続したソースコードの断片があったらそれを優先的に推薦する
 - プログラミングのロジック（例えばファイルがopenされたらいずれcloseされることを予測した推薦）を利用する

評価方法

- クロールしたソースコードの検索の性能を評価する.
- ソースコードの推薦がよりモダンであるかどうかを基準とし, 予め意図したソースコードが上位にくるかどうかで判断する.
- 例えばJavaScriptでは進化が早く, 一般的な検索(Googleなど)では古い情報が出てくることがしばしばある.

提案システムの構成図



今後の課題と予定

- ソースコードとメタ情報を正確に紐付ける手段
 - クロールしたソースコード中のコメント
 - クロールしていくサイトの記事中の解説
- 複数の解決方法があった時の順位付の手法
- データセットを効率よく集めていく方法
- 実験

