Python競技プログラミングサイトの設計と開発

尾有 栄光　　平塚 太一　　松本 貴裕　　高見 友幸

大阪電気通信大学 総合情報学部

キーワード：競技プログラミング，学習用Webサイト，Python

1.　はじめに

1.1 背景

現在，自身のプログラミングスキルの証明や高度なIT人材を採用する手段として，競技プログラミングが活用されている．競技プログラミングとは，与えられた課題を解くプログラムをいかに素早くかつ正確に作成するかを競うものである．

国内の代表的な競技プログラミングサイトとして，AtCoder[1]やyukicoder[2]などがある．

AtCoderは国内最大級のプログラミングサイトであり，毎週，定期的にコンテストを開催している．また，初心者向けと上級者向けのコンテストがそれぞれあり，自分のレベルに合ったコンテストに参加することができる．

yukicoderは競技プログラミングの練習を目的として運営されているサイトである．競技よりも学習に重点を置いた競技サイトとなっているため，ユーザー側が競技の問題を出題することもできる．

1.2 目的

競技プログラミングはプログラミングを駆使して競い合うため，自身のプログラミングスキルの向上に繋がるものである．よって，競技を通して自身の不足している知識を発見し，学習した後，再度挑戦するというサイクルは，当然のように思われる．しかし，既存の競技プログラミングサイトでは，競技のみで完結している．そこで，競技サイトと学習サイトが連携することで，競技で得た情報を学習サイトで活用し，学習で得た情報を競技サイトで活用することができるのではないかと考えた．本研究では，競技と学習の間で相互作用を引き起こす競技プログラミングサイトを提案する．

1.3 関連研究

本研究のように，競技サイトと学習サイトが連携した一つのコンテンツを開発した研究は，現状存在しないが，中村ら[3]のWOJや上野ら[4]のCLECなど，Webシステムを用いてプログラミング環境の支援システムを開発した事例は多数存在する．しかし，これらのシステムは，競技プログラミングとの関連性は無く，あくまで授業の支援を目的として設計されたシステムである．

2.　Pythonについて

Pythonとは，C言語などと比較して簡潔で分かりやすい文法が特徴的なプログラミング言語の一つである．近年，特に機械学習といった分野では必須とされる言語であり，競技プログラミングにも使用されている．また，世界的に大変人気のある言語でもある[5]．よって，有用性という観点から，本研究ではPythonの競技プログラミングサイトの設計と開発を行う．

3. システムの設計

3.1 システムの設計について

相互作用する例として，例えば競技サイトで初心者向けの問題を作成する際に，学習サイトの正答率などを参照することで，より適切な難易度の問題を作成できるだろう．また，学習サイトの解答時間を参照することで，競技の時間設定をより適切なものにできるだろう．

最終目標は，競技と学習が相互作用する一つのコンテンツ(図１)を完成させることであるが，本研究では，学習サイトの完成を目指した．

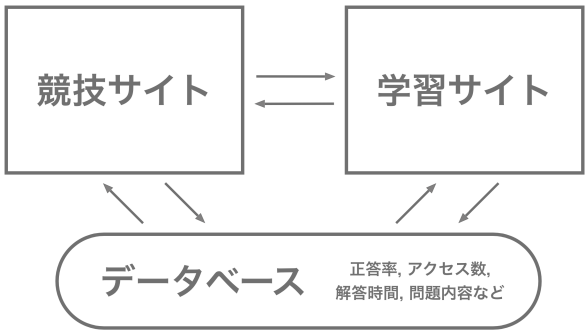

図１．システムの構成図.

3.2 開発環境

Webサイトの開発は，サイトのデザインやUIなどの見た目に関するフロントエンドと，データ処理やURLのルーティングを担当するバックエンドに分けられる．それぞれの開発環境について説明する．

3.2.1 フロントエンド

使用する言語はHTML，CSS，JavaScriptである．HTMLとCSSを用いてWebページにおけるデザインやUIを設計する．また，JavaScriptを用いてWeb上のボタンのクリックやスクロールなどのイベントに応じて処理を動的に変更する．このような開発手法は，Web開発においてはデファクトスタンダードであるため，本研究でも採用した．

3.2.2 バックエンド

使用する言語はPythonである．Web上で扱うデータを保持するデータベースは，Pythonに標準搭載で，別途サーバーを用意する必要がなく，扱いやすい，という理由でSQLite3を使用した．また，効率的に開発を行うために，FlaskというWeb開発フレームワークを使用した．Djangoという似たようなフレームワークも存在するが，Flaskの方が軽量で動作が軽いため，こちらを採用した．

3.3 システムの設計と開発

3.3.1 システムの設計

主にCheckiO[6]という既存のPython学習サイトを参考にして，システムの設計を行った．機能ごとに分割すると，ユーザー管理機能，学習機能，質問機能の３つの機能に大別される(図２)．それぞれの機能について解説する．

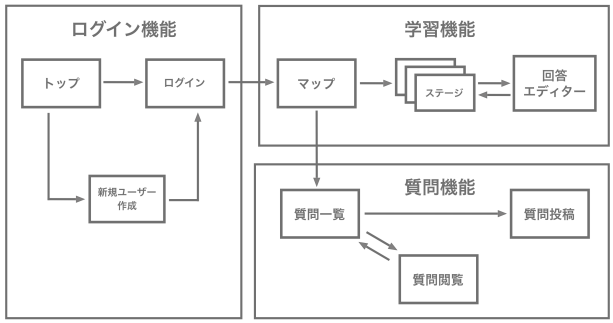

図２．学習サイトの状態遷移図.

3.3.2 ユーザー管理機能

ユーザー情報の管理を目的として設計した機能である．主にログイン処理を担当している．ログイン処理とは，ユーザーIDで個々のユーザーを識別し，パスワードでユーザーが本人であることを認証する仕組みである．また，同時にセッションIDという値を生成することで，ユーザーの状態を保持する役割も担う．

3.3.3 学習機能

ユーザーの学習を目的として設計した機能である．マップ(図２)で問題の種類ごとにステージという単位で分割されたものを選択できるようになっており，ユーザーはその中から自身が学習したい分野を選択する．次に，ユーザーはステージで問題を選択し，回答エディターで関数内に回答を入力する(図３)．その関数(関数名は問題に沿った名称)の出力値とデータベースにある回答値を照合することで正否を判定する仕組みになっている．

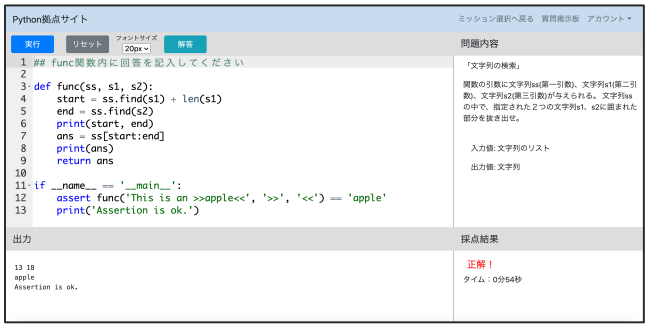

図３．回答エディター．

3.3.4 質問機能

ユーザーが学習する上で解決できない課題や，Pythonについての疑問を解消することを目的として設計した機能である．質問を投稿することで，質問一覧に掲示板のように表示される．質問をクリックすることで，質問閲覧ページに移り，質問を閲覧できる．質問閲覧ページ内にその質問に対する回答を投稿できる機能があり，他のユーザーが任意で回答できるシステムとなっている．

3.3.5 その他の機能

学習の結果を競技に反映するために，ユーザーのデータを管理する必要がある．そのため，前述の３つの機能の他にも，問題作成機能とデータ管理機能をそれぞれ実装した．これらは，管理者側の機能であるため，ユーザーは参照できない仕組みとなっている．

4. 回答エディターの実装

4.1 エディターの実装

ブラウザ上で動作するオープンソースのエディターで，CDNという技術を用いてWebサイト上にエディターを組み込むことができる．本研究では，このエディターを用いてエディター機能を実装した．

回答エディターには，出力の結果を確認する「実行ボタン」と実行時の出力値と解答の値が一致するか判定する「解答ボタン」がある(図３)．それぞれの機能について解説する．

4.2 実行処理

実行ボタンを押すことで，ユーザーのコードがuser.pyファイルに保存される．この際に，他のユーザーと競合が発生しないように，Pythonのfcntlモジュールで排他制御を行う．次に，Pythonのsubprocessモジュールで別プロセスとしてuser.pyを実行する．最後に，その出力結果をhtmlファイルに出力することで実行処理を実現している．

4.3 解答処理

解答ボタンを押すことで，htmlファイルのform要素の宛先を変更し，回答とは別のプロセスとして受け付ける．次にユーザーのコードがuser.pyファイルに保存される．そして，データベースから該当する問題の回答値を取得し，Pythonのassert構文を利用して正否を判定するassert文を生成する．最後に，Pythonの組み込み関数のexec()でそれらのassert文を順次実行し，その出力結果をhtmlファイルに出力することで解答処理を実現している．

5. 今後の課題

例えば，特定のファイルを参照することが，課題

の条件としてある場合，現状ファイルを扱う機能が無いため，そのようなファイル操作に関連する問題は実装できない．そのため，ユーザーがファイルを参照する仕組みや，問題で扱うファイルそのものを管理者が実装できる仕組みが必要である．

また，前述の通り，質問閲覧ページでは他のユーザーが任意に回答できる仕組みとなっている．しかし，回答者にメリットが無いため，十分な回答が得られない可能性がある．また，回答を他のユーザーに全て任せてしまっているため，質問がいつまで経っても回答されない可能性がある．そのため，例えばゲーミフィケーションを取り入れるなど，質問への回答を促す仕組みを導入する必要があると考えられる．

6. おわりに

本研究では，競技サイトと連携する学習サイトの開発を行った．ユーザーが問題を選択し，回答できるため，学習サイトとしては成り立っているが，前述のような問題があるため，引き続き学習サイトの開発を行う必要があるだろう．また，システムの設計で触れなかったが，運用するにあたっては，どのようにコンテンツをリリースしていくのかも検討していかなければならない．AWSのようなクラウド上に実装する場合でも，プラットフォームサービスを利用するのか，サーバーから構築していくのかで，実装方法は異なってくるからである．実装した後も，どのようにユーザーの情報を分析し，競技に反映していくのかについても考えていかなければならない．

実際にリリースして運用することで，少しでもユーザーが競技プログラミングで活躍できるようになることを期待する．

参考文献

[1]「AtCoder」
https://atcoder.jp，（参照2020年9月24日）．
[2]「yukicoder」
https://yukicoder.me，（参照2020年9月24日）．
[3]中村慎司，筧捷彦，“プログラミング授業システムWOJの開発”，情報処理学会第77回全国大会，Vol 4，939-940，2015.
[4]上野大貴，紫合治，“プログラミング学習環境のクラウド化とリモートペアプログラミング”，情報処理学会第76回全国大会，Vol4，553-554，2014.
[5]「The Top Programming Languages 2019-IEEE Spectrum」，https://spectrum.ieee.org/computing/software/the-top-programming-languages-2019（参照2020年9月19日）．
[6]「CheckiO」
https://py.checkio.org，（参照2020年9月24日）．