

hiki 編集作業を容易にするツールの開発

情報科学科 西谷研究室 3554 山根 亮太

1 目的

本研究を行う目的は、hiki の編集作業をより容易にするためのツールを開発することである。それを実現するために今回開発が必要と思われるところは hiki のコマンドである。具体的な作業は、hiki のコマンド名を shell 風にすることで瞬時にコマンドの機能を判断し使用することができるようにすることである。現在 hiki でコマンドを入力する際、ユーザーは瞬間的にそのコマンドの機能や意味を判断することが難しく迷いが生じてしまうことがある。しかし、この開発をすることでユーザーはコマンド入力の際にそのコマンドの機能が分からなくなって入力に迷ったり、コマンド自体を忘れていたりすることがなくなりより簡単に入力ができ、時間短縮にも繋がる。そのため、hiki のコマンド名を shell 風に書き換えることでこの問題を解決したい。

2 システムについて

2.1 hiki について

hiki とは wiki エンジンの 1 つとされており、プログラミング言語 Ruby を用いられることで作られた wiki クローンである。主な hiki の特徴として

- オリジナル wiki に似たシンプルな書式
- プラグインによる機能拡張
- 出力する HTML を柔軟に変更可能
- ページにカテゴリ付けできる
- CSS を使ったテーマ機能
- 携帯端末可能
- InterWiki のサポート
- HikiFarm に対応
- ページの追加、編集がしやすい

等がある。[1]

2.2 hikiutils について

hikiutils とは hiki の編集作業を容易に行うことができるようにするツール群のことである。よって、今回の研究は hikiutils のプログラムを編集することで瞬時にコマンドの機能や意味を理解できるコマンド名を考え編集を行う。hikiutils はプログラミング言語 Ruby のライブラリである gem によって提供されている。[2]hikiutils は CLI(Command Line Interface) で提供されているため、プログラミング言語 Ruby 版のコマンドラインのオプション解析を行いコマンドを標準化にする optparse が利用されている。[3] そのため、CLI の

optparse を簡単に扱うために Ruby にはいくつかの library が用意されている。

3 手法

Todo アプリケーションのプログラムの構造は lib/todo/command/options.rb にサブコマンド用の定義があり、入力されたコマンドは lib/todo/command.rb で判断されることにより lib/todo/command/options.rb 内で実行される。hikiutils のプログラムの構造は Todo アプリケーションのプログラムの構造と似たものであり hiki の編集を容易にするためのツール群のため、この部分の編集を行うことでコマンド名を書き換える。

4 考察

現在 Todo アプリケーションの作成を行うことでプログラムの構造を理解している。しかし、hiki のコマンド名の変更や直感的なコマンド名を考えることは現在できていない。今後の課題としては、まず初めに Todo アプリケーションの構造を理解することで hikiutils の構造を理解する。また、誰が見ても機能が判断できるコマンド名を考え、実際に hikiutils 内でそのコマンドの編集を行って実装し、そのコマンドが分かりやすいものであるかどうかを試してもらう必要がある。

参考文献

- [1] 「hiki」, hikiwiki.org/ja/about.html, 2016/9/2 アクセス。
- [2] 「Rubygems-Wikipedia」, ja.wikipedia.org/wiki/RubyGems, 2016/9/2 アクセス。
- [3] 「optparse の使い方」, dharry.hatenablog.com/entry/20081008/1223490673, 2016/9/2 アクセス。