

YagiWiki : コンテンツの作成・整理が簡単に行える Wiki の作成

YagiWiki : Interactive Contents Creation and Arrangement Wiki System

八木原 勇太 寺田 実
電気通信大学大学院 情報通信工学専攻
Yuuta YAGIHARA Minoru TERADA
University of Electro-Communications

概要

Wiki は、ブラウザ上から Web ページを作成することができるシステムで、個人的なメモ帳代わりとして、また様々な情報のまとめサイトとして広く利用されている。しかし既存の Wiki は、ページの再構成をしてコンテンツを整理するのが面倒であり、更新頻度が高い Web サイトでは扱いづらい。

そこで、コンテンツの整理整頓がしやすいように書き込みを短い単位で行い、それらを後で並べて配置することでページを構成する Wiki を作成した。これにより、思いついたことをより手軽に書き込むことができるようになり、「とりあえずメモ、後で整理整頓」という使い方が可能となった。

1 はじめに

1.1 Wiki について

Wiki というシステムが Web 上で広く利用されている。これはブラウザ上で Web ページを作成するもので、誰でも自由に書き込み・編集ができるのが特徴である。また複数人で別々のページを分担し、協力してコンテンツを作成することができる。Wiki は個人的なメモ帳代わりや、複数ユーザでの情報共有、また様々な情報のまとめサイトの製作などに利用されることが多い。しかし、既存の Wiki はいくつかの問題点を抱えている。

多くの Wiki は、コンテンツをページ単位で管理している。新しく何か書き込むときは、新規ページを作成してそこに書き込むか、既に存在するページに追記することになる。このため、書き込みの度に書き込む場所を考えなければならない、気軽に書き込みできない。また、一度書き込んだ内容を再構成するための手段が用意されていない。例えば大きなページを複数ページに分割したり、また複数のページをひとつに統合する等の作業は全て手作業で行わなければならないため、コンテンツの整理に多くの労力を要する。

1.2 ストック型の情報とフロー型の情報

情報にはストック (stock) 型の情報とフロー (flow) 型の情報がある。ストック型の情報とは、ものごとを系統立ててまとめて蓄えられた情報のことで、フロー型の情報は、外部から次々と入ってくる新しい情報である。多くの Wiki はストック型の情報、Blog 等のツールはフロー型の情報を扱うのに適している。

多くの Wiki は、新規書き込みの問題から、フロー型の情報をうまく扱うことができないとされている。また、ページの分割・統合の問題点から、ストック型の情報についても十分に扱えていないと考えられる。

1.3 方針

本研究では既存の Wiki の利便性を生かしつつ、先に述べた新規書き込みとページの分割・統合の問題を改善し、コンテンツをより柔軟に扱えるシステムの構築を考える。

そこで本システムでは、文章の書き込みを短い単位で行い、それらを後で並べて配置することで、ページの整理整頓を行う。つまり、新しく入ってきたフロー情報を一つ一つバラバラに蓄えておき、それらを後で整理、また必要に応じて情報のリファクタリングを行うことでストック情報として保持することができる。

2 既存システム

2.1 KJ 法

KJ 法とは、川喜田二郎が考案した情報整理の手法であり、大量の情報をまとめて整理したり発想支援の補助として用いられている。文章の要約や収集したデータ等を小さなカードに書き、それらをいくつかのグループに分け、それらを平面に配置することで情報を整理する。

文章を細かい単位で分けて書き、それらを並べることでコンテンツを製作するという点では、KJ 法は本研究と類似する。しかし、KJ 法は多くの情報をまとめる過程でカードを利用するだけであり、カードの配置結果を

ソフトウェア

Mozilla Firefox

Mozilla Japan - Firefox
<http://www.mozilla-japan.org/products/firefox/>

Mozilla Firefox プロジェクトにより開発されているオープンソースな Web ブラウザ。デフォルトではやや低機能だが、拡張を入れることでいろいろな機能が追加できる。

ThumbSense

ThumbSense
<http://www.sonycsl.co.jp/person/rekimoto/tsense/>

ノート PC のタッチパッド + キーでマウスクリック等の操作ができる。ウィンドウを閉じたり移動、最大化したりすることも可能。手をホームポジションから動かさずに操作できるので、慣れるとマウスよりも素早く作業できる。

bluewind

bluewind

図 1: 文章を細かい単位で区切り、それらを並べる

そのまま保存して利用することはない。本システムでは、文を配置したものがそのまま Web ページとして閲覧可能になり、さらに新しい文章を追加するなどの編集も可能だ。つまり配置作業を繰り返すことで常に更新し続けることができる。

頻繁に更新される Web ページの場合、一度作成したコンテンツに新たに情報を追記したり、またそれによってページの構成が変わることがあるが、KJ 法ではこのような常に更新され続ける文章を作るためには設計されていない。つまり、KJ 法はストック型のコンテンツを作成するための手法としては有効だが、新しく入ってくるフロー型の情報に対応できない。

2.2 Blog

Blog は、日記のように新しいコンテンツをどんどん書き加えていく形式の Web ページを作成するツールである。Web ブラウザや携帯電話から気軽に書き込むことができる。用途としては、日記やニュース記事に対する意見を書く以外にも、新たに知ったことや気になったこと等を備忘録としてメモする用途でも利用されている。

しかし、Blog は基本的にフロー型の情報のみを扱うものであり、書き込んだ記事 (エントリ) は書きっぱなしのまま蓄積されていく。後で書き込んだ内容を参照し

たり整理したりすることがほとんど無いため、大量のコンテンツをまとめるには不向きであると考える。

2.3 wema

wema は、付箋紙を並べて Web ページを構成するシステムである。短い文章を付箋紙に書いていき、それらを二次元平面上に並べることができる。また、Wiki と同様に誰でも編集・削除が可能のため、多人数で情報共有に使用することもできる。

wema は紙の上に字を書くように自由にメモなどを記述できるため、新しい情報をどんどん書き込むフロー型のシステムとしては有効だが、情報を系統立ててまとめるストック型のシステムとしては不向きである。また、更新頻度の高いページの場合、頻繁にページ全体の再構成が必要となり、収集がつかなくなることがある。

2.4 PositLog

PositLog は、Web ページ上に直接文字や絵を書き込んだりして、絵地図のようなページを作ることができるシステムである。

wema と同様、二次元平面状で自由にページレイアウトすることが可能であり、フロー型の情報整理に適しているが、量が多くなったときに煩雑になりやすく、一度煩雑になったページから系統立てた見やすいページを作成するのは困難である。

2.5 TiddlyWiki

TiddlyWiki は、文章をページ単位ではなく、パラグラフ (Tiddler) 単位で扱う Wiki である。

通常の Wiki では、ページ内のリンクをクリックするとページ遷移が起こり、今まで見ていたページは見えなくなってしまう。TiddlyWiki では、リンクをクリックすると新しいパラグラフがリンクのあるパラグラフの下に現れ、同じページ内に並んで表示される。また、表示されているパラグラフをそのまま編集して更新することができるため、パラグラフ間のコンテンツの移動などの再構成が容易に行える。しかし、パラグラフの表示順は閲覧時に決まるため、コンテンツ作成者側からは表示を制御できず、まとまったコンテンツは作りづらい。

3 システムのモデル

本システムでは、コンテンツを後で整理しやすくするために、コンテンツの書き込みを全て 50~100 字程度の短い単位で行う。それにより、思いついたことを手軽に書き込むことができ、また後に整理整頓を簡単に行うこ

とができるようになる。

本システムは断片・ページ・ディレクトリの3つの要素で構成される(図2)。

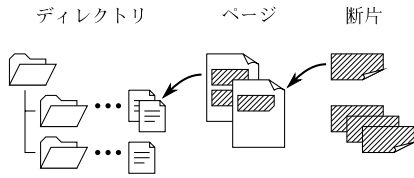


図2: YagiWikiのドキュメントモデル

3.1 断片

断片とは、ユーザによる書き込みの一単位を指す。何かあったときにとりあえずメモしておく、という使い方が考えられるため、これをメモ帳のような小さい紙の端切れに書いていると考え、本稿ではこれを断片と呼ぶ。

各断片には、タイトル・検索性キーワード・スタイルを設定することができ、作成日時・更新日時とあわせてWebサーバ上に記録される。スタイルはテキスト・整形済みテキスト・箇条書きの3つがあり、Wiki文法のような特殊記号を用いずにこれらを記述することができる。また、断片を作成すると、作成した月ごとに新しいページが自動生成され、日付単位でセクションにまとめて配置される。

断片は、断片を書き込む専用の書き込みページから作成するが、ページ編集時にも作成することができる。

また、ひとつの断片を複数のページに配置することができる。これを断片の共有と呼ぶ。このとき、片方の断片を編集したときにもう片方の断片も連動して更新される。

3.2 ページ

ページとは、既存のWikiのページに(ウィキページ)に相当するもので、ページ内に複数の断片を一行に並べて構成する。これは、大きな白紙の上に断片をひとつずつ貼り付けていくようなイメージである。

各ページにはタイトルを設定することができ、作成日時・更新日時とあわせて記録される。

ページを編集するときはページ編集モードに切り替え、マウス操作により断片を配置する。

ラベル

ページには断片を並べるだけでなく、直接文字を書くこともできる。本稿ではこれをラベルと呼ぶ。

ラベルは、並べた断片に補足説明を加えるように貼り付けたり、小さな見出しとして利用することができる。

セクション構造

多くのWikiではWiki文法を用いて「見出し」を作成しページ内に階層構造を作れるが、本システムではセクション構造という形で階層構造を作る。

ひとつのセクションの中に関連のある断片をまとめて配置することで、大量の断片を簡単に整理することが可能となる。また、セクションの中にセクションを入れ子にすることもできるため、階層構造をもった文章を簡単に記述することができる。

3.3 ディレクトリ

複数のページを整理するために、任意にディレクトリ構造を作ることができる。これはWebページを作成する際に、関連する内容のページをまとめてひとつのディレクトリの中に配置することと同じである。

ページ管理モードにてディレクトリを作成したり、作成したページを配置したりすることができる。

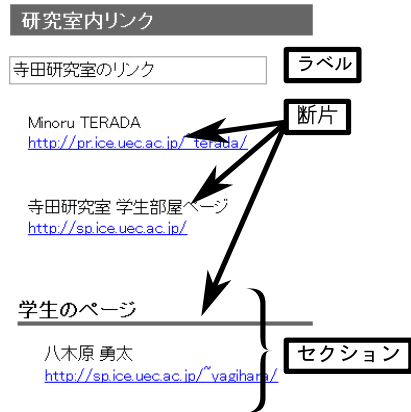


図3: ラベル・断片・セクション構造の表示

3.4 ドキュメントモデルの妥当性

断片という単位の導入について

既存のWikiの問題点として、一度書き込んだ内容の再構成は全て手作業で行わなければならない。ウィキページを実際に再構成する際は、作成したパラグラフご

とに配置するページを決め、カット & ペーストによってコンテンツを移動する。これをできるだけ簡単に行えるようにするため、コンテンツを扱うページよりも小さい「断片」という単位を導入し、それらをページに配置することで構成することを考えた。断片をドラッグ & ドロップで移動できるようにすることで、ページ内で断片の並び順を変えたり、ページ間で断片を移動することが容易になり、ページの再構成が行いやすくなると考えられる。

また、コンテンツの新規書き込みの度に書き込む場所を考えなければならないという問題点がある。これに対しては、新しい情報を全て断片として書き溜めていくことで、何も気にせず書き込みを行うことができるようになる。断片の量が多くなったときは前述のようにドラッグ & ドロップによりページの再構成を行うことができる。

つまり、新しいフロー型の情報は断片を作成して書き残しておくことで書き留めておき、それらを再構成するなどの整理整頓をしやすいしてストック型の情報として蓄積できる。

既存手法の問題点

FreeStyleWiki [5], MediaWiki [6] 等、いくつかの Wiki はパラグラフ単位で書き込み内容を編集する機能をもっている。これにより、コンテンツをページよりも細かい単位で扱うことができるが、この機能だけでは書き込み内容を別のページに移動することができないため、ページの再構成に関しては既存の Wiki と同様、扱いづらい。

また PukiWiki [3], FreeStyleWiki [5] などには、あるページを別のページの中に挿入する機能がある。この機能を用いることで、Wiki のコンテンツを柔軟に扱うことができるが、常に更新され続けるコンテンツを扱うには不十分である。例えば、あるページに少しずつコンテンツを追記していき、ある程度大きくなったときにページを分割、といった作業の手間は、この機能だけでは解決できない。

4 利用例

4.1 とりあえずメモ、後で整理整頓

本システムは、新しいフロー型の情報を断片として残しておいて、後でページに配置することで整理を行う、という使い方ができる。断片の移動はドラッグ & ドロップでできるため、容易にページの整理整頓ができる (図 4)。本節ではそのような利用例を紹介する。

CUI プログラムのコマンドをメモ

CUI ベースのプログラムやシステムのコマンドを、新しく覚えたものをどんどん断片として書き溜めていくという利用法が考えられる。断片をいくつか作成した後、ページ編集モードに切り替えて、作成した断片を並べることでページを作ることができる。断片は、作成した月ごとに自動生成されたページに配置されているため、断片を作成した月のページを開いて探すことになる。また、検索機能を用いて指定したキーワードにマッチする断片をリストアップし、そこから以前作成した断片を探すこともできる。

このようにして作成したページが多くなった場合は、ディレクトリ編集モードに切り替え、好きなようにディレクトリ構成を変更することができる。

convert コマンド

形式変換

```
$ convert infile.png outfile.eps
```

画像加工

サイズ指定

```
$ convert -geometry 320x240 infile.png outfile.png
```

画像の回転

```
$ convert -rotate 90 infile.png outfile.png
```

透過処理

```
$ convert -t #ffffff infile.png outfile.png
```

減色処理

図 5: プログラムコマンドのメモ

リンク集を作る

新しい Web サイトを見つけたときに、Web ページのタイトルと URL をひとつの断片に記録して蓄えておく。断片がいくつかできたところで、検索機能を用いて 'http' 等のキーワードで検索すると、蓄えた断片が全て列挙される。これをそのまま編集し、ページタイトルをつけて新規ページを作成することができる。ここで作成したページも同様に、ディレクトリ編集モードで好きなディレクトリに配置することができる。

また、同じ断片を複数のページで共有する機能を用い

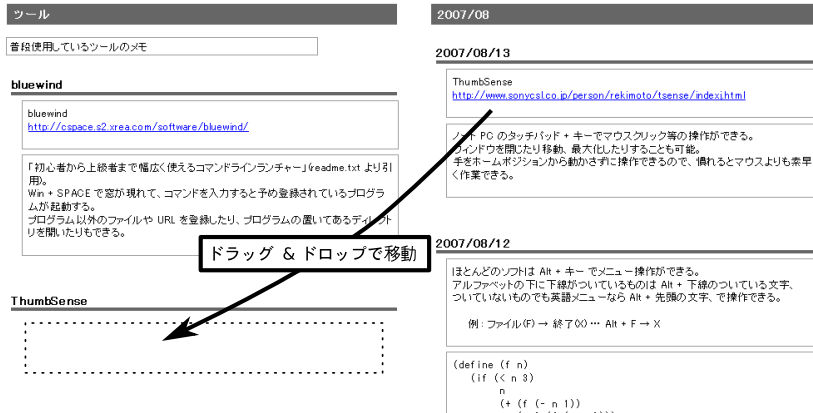


図 4: ドラッグ & ドロップによる編集



図 6: 断片の検索 ('http' で検索した例)

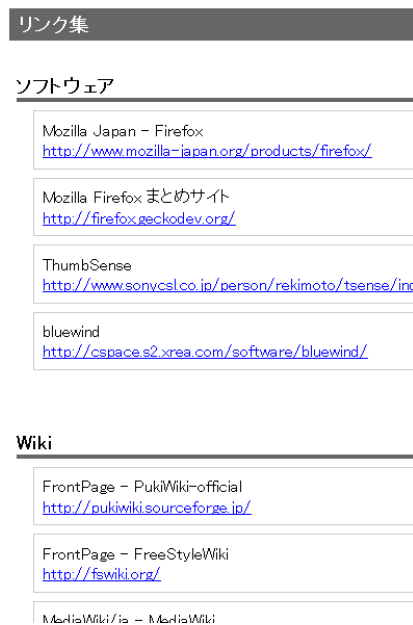


図 7: 作成したリンクページの例

ることで、同じページへのリンクを複数のページに設置することができる。このとき、例えば Web サイトの移転等で URL に変更があった場合、共有した断片のうちひとつを書き換えることで他の断片も連動して更新することができる。

4.2 既存の Wiki と同様の使い方

本システムでは、ページ編集モードでも新規断片をページ中に追加することができるため、既存の Wiki と同様の編集方法もできる。このような使い方は、作成するページの構成がある程度決まっているときに有効である。

プログラム言語のまとめページ

プログラミング言語を系統立ててまとめたページを作成するときなどは、既存の Wiki のような使い方で作成することができる。この場合は、新たに新規ページを作成し、そこに新規断片を作成しながら配置する。



図 8: C 言語まとめページの作成

4.3 リファクタリング

本システムでは、複数ページを横に並べて表示したり、ページ間でコンテンツを移動することができるため、大きくなったページを複数のページに分割したり、複数のページをひとつのページに統合したりする作業を容易に行える。例えば、ページのコンテンツの一部を別のページに移動する処理は次のような流れになる。

まず編集するページとコンテンツの移動先のページを並べて表示する。これはリンクの横に表示されるポップアップメニューから行うことができる。

次に、編集モードに切り替え、コンテンツを移動する。コンテンツの移動は、断片単位もしくはセクション単位で行い、いずれもドラッグ & ドロップで移動することができる。複数断片をまとめて移動することができ、また関連のある断片をひとつのセクションにまとめておくことで、複数のコンテンツをまとめて別のページに移動することが容易にできる (図 9)。

5 評価実験

本システムを、6 人の被験者に約 1 ヶ月間利用してもらい、利用状況から気づいた点をいくつか考察した。

5.1 断片の書き込み

断片の書き込みは、書き込みページ上のフォームからとページ編集モードからの 2 通りがあるが、今回の実験では多くの断片がページ編集モードで作成されていた。これは、既に作成されたページに追記する書き込みが多く、断片作成と同時にページに配置するためにページ編集モードで作成されたと考えられる。フォームから作成された断片は、断片作成の時点で適切なページがまだ作成されておらず、「とりあえずメモしておいた」というものが多かった。

5.2 断片のサイズ

今回の実験で、ひとつの断片に大量の文章を書き込み、整理がしにくくなっている例が見受けられた。それらのほとんどは、見出しの文字列を断片の中に含めてしまったり、主題の異なる内容の文章を同じ断片にまとめて書いてしまうものであった。

本システムでは、長い文が書かれた断片を複数の断片に分けたり、逆に複数の断片をひとつの断片に統合するには、現段階では手作業で断片の編集を行わなければならない。これを、どのような UI で実装したらよいか、今後の研究課題となるだろう。



図 9: 断片・セクションのページ間移動

5.3 断片の共有

同じ断片を複数のページに貼り付ける、断片の共有が数箇所で行われていた。中でも、ひとつの断片に Web ページのタイトルと URL を書いたブックマークとしての利用法は、本実験の中で考案されたものである。Web サイトの移転により URL が変更された場合、ひとつの断片を修正すると他の全ての断片に反映されるため、非常に効率が良い。

6 今後の課題

6.1 コンテンツ整理を支援する機能の実装

断片作成時の自動配置

断片を作成すると、作成した月ごとにページが自動的に生成され、そこに自動配置される。ここで、キーワード等を指定して、断片の作成と同時にユーザが指定したページに自動的に配置するような機能を実装したい。具体的には、例えば「latex」を文中に含む断片が新規作成されたら、「LaTeX ページ」の指定セクションの中に自動的に配置されるような機能である。

このような機能的は、システムの使い方等をまとめたページを作成する際、未整理情報をまとめておくのに役に立つと考えられる。

ローカルファイルから断片の作成

移動中などでネットワークが利用できない場合等、一時的なメモをテキストファイルとしてローカルに保存しておくことがある。このようなファイルをまとめてアップロードし、断片を一括して作成する機能を実装したい。

このとき、長いテキストを編集しやすいよう短い単位

で区切り、自動的に複数の断片に分割する処理が必要となる。テキストを区切る基準や、ユーザにとってわかりやすい分割の指定方法を考えなければならない。

断片の分割・統合

現段階では、長い断片を複数の断片に分割したり短い断片をひとつの断片に統合したりする処理は、全て手作業で行わなければならない。「とりあえずメモ、後で整理整頓」という使い方を実現するためには、断片の大きさなど、余計なことを気にせず書き込みができるように、編集時に簡単に断片の分割・統合が行えるようにするのが望ましい。これを初心者にもわかりやすく簡単にできるようなインターフェースを考え、実装したい。

ページ分割の支援

ページの分割は現在は断片やセクションをマウス操作でひとつひとつ移動しなければならない。既存の Wiki のコピー & ペーストと比較すると作業量は大幅に改善されているが、よりユーザ負荷の少ない手法を考え、実装したい。

例えば、ページ内に配置されているセクションのうち最もレベルの高いものをそれぞれ新しいページとして作成する、といった作業はシステム側で自動的に行うことができる。このような処理を行うためのインターフェース等を考え、実装したい。

6.2 具体的なタスクによる評価実験

今回は、特に内容を定めずに自由に利用してもらうという形式で評価実験を実施した。そのため、本システムの情報整理的な部分を十分に評価することができな

かった。

今後は、具体的にどのようなページを作成するかタスクを決めて、どの程度効率よく情報整理が行えるかを評価したい。また、多くの情報をまとめるストック型の情報と、新しく入ってくるフロー型の情報の両方をどの程度効率よく扱えるか、今後の研究課題である。

参考文献

- [1] KJ 法本部・川喜田研究所 - KJ 法/KJ Method.
<http://www.kj-method.jp/> .
- [2] KJ 法
<http://nokai.ab-garden.ehdo.go.jp/giho/43.shtml> .
- [3] PukiWiki-official
<http://pukiwiki.sourceforge.jp/> .
- [4] YukiWiki
<http://www.hyuki.com/yukiwiki/> .
- [5] FreeStyleWiki
<http://fswiki.org/> .
- [6] MediaWiki/ja
<http://www.mediawiki.org/wiki/MediaWiki/ja> .
- [7] ブログ - Wikipedia
<http://ja.wikipedia.org/wiki/%E3%83%96%E3%83%AD%E3%82%B0> .
- [8] wema - 付箋指向の多目的ツール
<http://wema.sourceforge.jp/> .
- [9] PositLog [ポジログ]
<http://positlog.storybook.jp/> .
- [10] TiddlyWiki
<http://www.tiddlywiki.com/> .