

表データ操作をRDBで強化したWikiシステム

奥村俊也^{†1} 寺田 実^{†1} 丸山 一貴^{†2}

従来の Wiki は誰でも気軽に編集できる反面、膨大なデータを扱うにはそれに比例した文章量の編集が必要になる。一方、RDB は表データを管理する際に非常に有用ではあるが、SQL は一般的な Wiki ユーザには敷居が高い。

本研究では、Wiki 記法による RDB へのデータ入力と、拡張した Wiki 記法によるデータ表示を提案する。それにより、表データ操作が容易な RDB インタフェースを持つ Wiki システムを実現した。

A Wiki System Enhanced by Visible RDB Operations

TOSHIYA OKUMURA,^{†1} MINORU TERADA ^{†1}
and KAZUTAKA MARUYAMA^{†2}

Although Wiki systems can easily be edited by everyone, the more data we treat, the more sentences must be needed to edit on usual Wiki Systems. On the other hand, RDB Systems are very useful when the table data is managed, but it is not made for general wiki users.

In this paper, we proposes the representation for RDB operations by the enhanced Wiki notation. As a result, the new Wiki has the RDB interface which makes the manipulation of table data easier.

^{†1} 電気通信大学電気通信学研究科情報通信工学専攻

Graduate School of Information and Communication Engineering, The University of Electro-Communications

^{†2} 東京大学 情報基盤センター

Information Technology Center, The University of Tokyo

1. はじめに

近年見られる情報爆発により、現在 Web 上には膨大な量の情報が存在している。そのため、複数のユーザが様々な情報を持ち寄り、情報が収集、整理される場所が必要となる。

Wiki システムを活用した Web 上のフリー百科事典である Wikipedia¹²⁾ には多数の記事が存在し、内容が活発に編集されていることから、Wiki システムは、情報を一つにまとめて有用なコンテンツを構築する場合に適していることが分かる。しかし、多種多様な情報が一箇所に集まることにより、編集頻度と情報量は更に増えるため、情報の整理を行い、コンテンツ内容の可読性を高める必要がある。

一般的に、テーブル構造を用いて様々な情報を管理、提示することは非常に有効である。ところが、従来の Wiki において大量の情報を編集するには、それ以上の量の文章を編集する必要があり、編集作業が煩雑になってしまう。あるユーザが所持する特定コンテンツに関しての情報量と、そのユーザの編集スキルは必ずしも比例しているわけではないため、情報量がある程度の規模になった時点で編集の煩雑性によって編集ユーザが減少し、情報の提供が止まってしまう恐れがある。そこで、テーブル構造のデータをより容易に編集、操作できる RDB システムに着目した。

本研究では、Wiki と RDB に着目し、存在する膨大な情報を表データを用いて Wiki 上で編集、管理を行うことで、ユーザが情報を容易に探索できる新しいインタフェースの提案を行う。

1.1 Wiki システム上での情報整理

Wiki システムとは、ウォード・カニンガム氏が発明した、Web ブラウザからサーバ上の文書を複数ユーザが編集できる CMS(コンテンツマネージメントシステム) の一つである¹⁾。誰でも、Web 環境があればどこからでも閲覧、編集が可能であり、多人数による情報整理に有用である。

しかし、取り扱うデータ量が膨大になるにつれ、編集時に扱う文章量が肥大化し、編集作業が煩雑になってしまう恐れがある。

1.2 RDB 上での情報整理

RDB(リレーショナルデータベース)とは、テーブルと呼ばれる表構造を用いてデータを管理し、複数のデータの結合や抽出を容易に行うことができるシステムである。膨大な量の情報を扱う場合、RDB を用いることで安定かつ高速にデータ管理を行うことができる。

しかし、システムを利用するには、まず SQL 文と呼ばれる専用の操作言語を習得する必

要があり、一般的な Wiki ユーザが利用することは難しいと考えられる。また、RDB 単体ではデータを管理する手段としてテーブル構造を用いることしかできず、Web 上に表示させるには、別途システムを構築し、そのシステムを通して操作する必要があるため、非常に手間がかかる。

2. 既存システム

前述の問題点を改善するため、様々な機能を追加した新規 Web システムが提案されてきた。Wiki においては、特定の機能を特化させ、用途に合わせた独自 Wiki システムが開発されている¹⁾²⁾³⁾。また、RDB は安定性と高速性に注目され、早くから WWW と大規模 RDB の連携システムが研究、開発されている⁴⁾⁵⁾。ここではその一例を紹介する。

2.1 PukiWiki と PukiWiki プラグイン

PukiWiki⁷⁾ は、PHP を用いた Wiki システムである。Wiki ページをカテゴリ別に分類できるなどの情報管理に特化した機能の他、プラグイン形式で様々な機能を付加でき、拡張性も高い。実際に欲しい機能があれば、プラグインを自作することで実装することもできる。PukiWiki の公式サイト⁷⁾ には、自作したプラグインの公開や機能の提案を行う場も設けられており、機能の更新や追加が活発に行われている。

2.1.1 areaedit プラグイン

areaedit.inc.php⁸⁾ は、ページ内の指定された範囲のみを編集対象にできる PukiWiki プラグインである。ページ編集時に範囲を指定することでページ閲覧時に編集リンクが出現し、そのリンクをクリックすることでその範囲のみを編集することができる(図 1)。これにより 1 ページあたりの情報量が肥大化しても、予め編集範囲を細分化でき、膨大な文章を編集する必要はなくなる。編集の煩雑性が低くなるだけでなく、編集失敗時の修正も容易であり、非常に有用なプラグインである。また、Wiki 編集に慣れていない一般ユーザが簡単に編集できるような範囲指定を行うことで、副次的に編集ユーザの区別がなされると考えられる。

2.1.2 table_edit プラグイン

table_edit.inc.php⁹⁾ は、Wiki 上における表データの操作を容易に行う PukiWiki プラグインである。表データを編集する際に引数を指定することで、特定の列をキーとして行をソートさせた表データを表示させることができる(図 2)。また、行ごとにリンクが自動的に設置され、クリックすることでその行のセル内容のみを簡単に編集することができる。

従来の Wiki では表データの編集は編集画面上において手作業でコピーアンドペーストを用いて行う手法が一般的であった。しかし、その方法では編集ミスが多発し、表の構造自体

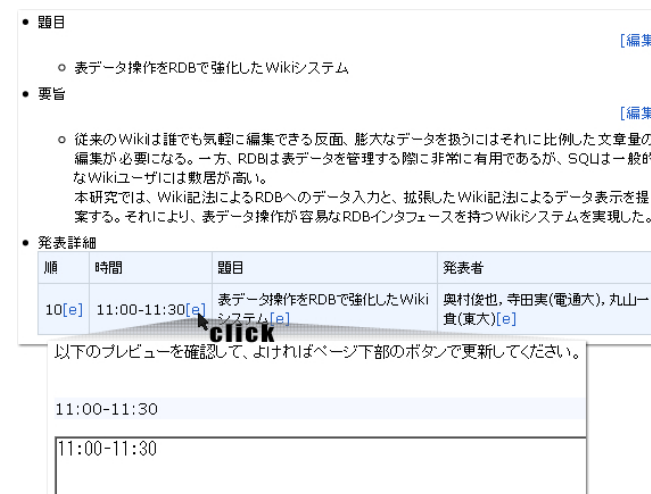


図 1 areaedit.inc.php⁸⁾
“[編集]”もしくは“[e]”というリンクをクリックすると、その範囲のみの編集フォームが表示される。

が変化してしまう恐れがある。その可能性を回避し、引数のみでソートが行える点は、膨大なデータを操作する際にも有用であると考えられる。

2.1.3 PukiWiki プラグインにおける着眼点

これらのプラグインを用いることで、Wiki 上における表データの編集作業は容易になる。しかし、RDB のように完成された表データに対しての様々な操作を行うことができない。ページ内における情報の可読性を更に高めるには、表データの特定列、特定行のみの表示や、複数の表データの結合など、表データの RDB 操作が必要である。

本研究においては、本プラグインを、表データ自体を編集する機能の実装において参考にした。

2.2 YagiWiki

YagiWiki⁶⁾ は、断片と呼ばれる短い単位のドキュメントを用いることにより、情報の整理に特化した独自の Wiki システムである(図 3)。マウスのドラッグ操作を行い、断片を自由に並べて配置することによってページを構成する。入力されたデータ量が膨大になった場合も、そのデータを整理する操作は非常に容易である。更に、断片は複数のページ間で共有することが可能である。一つの断片を編集すると、その断片を配置している複数のページで

期	時間	題目	発表者
11	11:30-12:00	両手協応操作型インタフェース～指示特操作に関する実験システムの開発～	多胡尚(早大)
12	12:00-12:30	多画面対応ZUI議論ソフトウェアBorderless Canvas	栗原一貴(産総研)
10	11:00-11:30	表データ操作をRDBで強化したWikiシステム	奥村俊也(電通大)

table1

-- 題目
-- 表データ操作をRDBで強化したWikiシステム
-- 要旨
-- 従来のWikiは誰でも気軽に編集できる反面、膨大なデータを扱うにはそれに比例した文章量の編集が必要になる。一方、RDBは表データを管理する際に非常に有用であるが、SQLは一般的なWikiユーザには敷居が高い。––
本研究では、Wiki記法によるRDBへのデータ入力と、拡張したWiki記法によるデータ表示を提案する。それにより、表データ操作が容易なRDBインタフェースを持つWikiシステムを実現した。
-- 発表詳細

#table_edit(table1, 1n) table1:表名 1n:1列目を数値ソート

- 題目
 - 表データ操作をRDBで強化したWikiシステム
- 要旨
 - 従来のWikiは誰でも気軽に編集できる反面、膨大なデータを扱うにはそれに比例した文章量の編集が必要になる。一方、RDBは表データを管理する際に非常に有用であるが、SQLは一般的なWikiユーザには敷居が高い。本研究では、Wiki記法によるRDBへのデータ入力と、拡張したWiki記法によるデータ表示を提案する。それにより、表データ操作が容易なRDBインタフェースを持つWikiシステムを実現した。
- 発表詳細

期	時間	題目	発表者
10	11:00-11:30	表データ操作をRDBで強化したWikiシステム	奥村俊也(電通大)
11	11:30-12:00	両手協応操作型インタフェース～指示特操作に関する実験システムの開発～	多胡尚(早大)
12	12:00-12:30	多画面対応ZUI議論ソフトウェアBorderless Canvas	栗原一貴(産総研)

図 2 table_edit.inc.php⁸⁾
1 列目を数値ソートした表 table1 をページ内に表示している。

その編集が反映されるため、編集の手間が軽減される。

2.2.1 YagiWiki における着眼点

断片自体を操作し、可読性を高めるとい点に関しては、既に存在するデータの操作が容易にできるため、情報量の多いページの編集煩雑性に対して非常に有用である。しかし、YagiWiki では表データに対しては特別な操作を行うことができないため、表データを構築し、ユーザに適当な情報を提示することは難しい。表データに RDB 操作を行うことができれば、ページ内の情報を操作する幅が更に広がり、情報管理に適したシステムとなる。

2.3 MySQL と phpMyAdmin

MySQL¹⁰⁾ は、RDB を管理運用するためのシステムである RDBMS の実装の一つである。オープンソース・データベースであり、他の RDBMS に比べて高速性に定評があるため、高頻度で参照されるデータを格納するアプリケーションに使用される傾向がある。

MySQL 内のデータを操作する管理ツールとして、ここでは phpMyAdmin¹¹⁾ を紹介する。phpMyAdmin は、PHP で実装されており、MySQL をインターネットを介してブラ

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。
このソフトは、タッチパッド + キーでマウスクリック等の操作ができる。

図 3 YagiWiki⁶⁾
枠で囲まれた部分が断片である。

ウザ上で管理することができる。

RDB を操作するには、特有の記法である SQL 文を入力する必要があるが、記法の複雑性が SQL でデータ管理を行う際の一つのネックになっていることは間違いでない。それに対して、phpMyAdmin では SQL 文の記述を必要とせずに MySQL 内のデータベースに対して操作を行うことができるため、その欠点を補うことに成功している。また、SQL 文を入力してデータを管理することも可能であり、単純にブラウザ上から MySQL にアクセスできるツールとしても使用可能である。

2.3.1 MySQL と phpMyAdmin における着眼点

phpMyAdmin は、MySQL の内容を容易に操作することができるツールであり、このままではユーザにデータを提示することができない。整理した情報を提示するには、別途に提示システムを構築する必要がある。また、RDB 単体ではテーブル構造しか扱うことができないため、例外処理を行うことができない。以上のように柔軟性が乏しく、ユーザが思い描いた編集を行うことは難しい。

3. 方針

Wiki システムの優れた操作性と、RDB の高いデータ管理能力を持ち合わせた新しいシステムを提案する。

Wiki システムの課題である膨大なデータの取り扱いを RDB に任せ、編集や表示を Wiki 上で行うようにする。これにより、従来の Wiki では手間がかかっていた表データの編集や検索、ソートなどの操作を、これまでの Wiki 記法と似た独自記法を使って Wiki 上で容易かつ高速に行うことができ、表データ操作を強化した Wiki システムとして利用できる。

また、RDB からの視点でこのシステムを考えた場合は、SQL 文の理解などの必要なスキルの高さ、表データ構造のみにしか対応していないなどの柔軟性の低さ、といった問題点を解決している。編集は全て簡素化した独自の Wiki 記法を用いるため、SQL 文を用いる必要がなく、RDB の知識を持っていないユーザも扱うことが可能である。表示は全て Wiki 上で行うため、表データの前後にアノテーションとしてドキュメントを挿入することもできる。

以上のように、Wiki、RDB 両システムの利点を最大限に活かしつつ、大量のデータを扱う際に発生する問題点を補ったシステムを構築した。その結果、膨大な情報をユーザが思い描いた通りに管理でき、汎用性が高いシステムとなった。

4. システムモデル

本研究では、前章で述べた方針を軸に、既存システムを参考にしながら PHP+MySQL を用いて Wiki システムの実装を行う。

Wiki 内のコンテンツをページデータと表データに分け、どちらも RDB に格納する (図 4)。表データの内容や構造の編集と、ページ内の文章や構造の編集を別ページにて行うため、効率の良い編集を行うことができる。

4.1 表データの編集ページにおける操作

4.1.1 表データの作成

表データは一つのページとして作成する。これにより、表データのみを編集することができ、通常テキストを編集する場合と区別化が図れるため、煩雑性が減少する。表データの作成には従来の Wiki と同じ “|” を用いた記法を用い (図 5)、Wiki ユーザが従来通りに表データを作成できるようにする。保存ボタンを押すことにより、記述されたドキュメントが表データとして自動的に RDB に格納される。

本来 RDB に格納する際には、単純なテーブル構造のデータだけでなく、フィールドの名

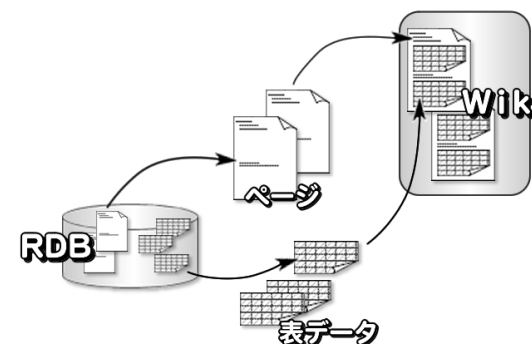


図 4 システム概要
ページ情報だけでなく、表データ自体も RDB に格納する。



図 5 表データを作成、編集した後、保存した。

前やデータの種類の指定する必要がある。従来の Wiki 記法における表データにはそれらが欠落している為、格納する際に補う必要がある。フィールドの名前は、指定されなければ汎用フィールド名 “title[num]” にし、表示時には非表示にする。データの種類の、デフォルト値を “text” とするが、例外としてそのフィールドに属するセル内のデータが全て数値であった場合に限り、“double” とする。これにより後述のフィールドをキーとした行ソートが可能になる。

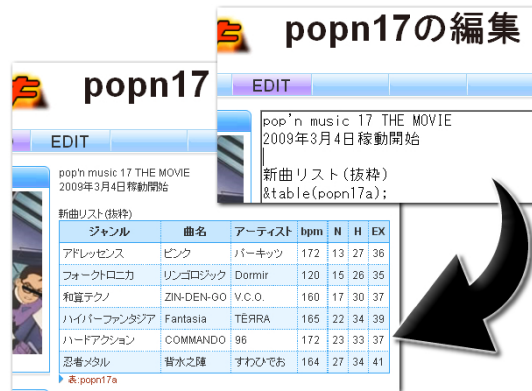


図 6 表データをそのまま表示した .



図 7 フィールド「ジャンル」「曲名」「EX」のみを表示した .

4.1.2 表データの編集

既に存在する表データを編集する際には、RDB から表データを読み込み、Wiki 記法に変換してドキュメントとしてユーザに提示する (図 5) . ユーザは、従来の Wiki と同じように Wiki 記法に従って表データの編集を行う . 編集が完了した時点で保存ボタンを押すことにより、編集された表データが RDB に格納される .

4.2 ページデータの編集ページにおける操作

表データ以外の文章を従来の Wiki 記法を用いて編集を行う . 表データをページ内に表示させたい場合は、以下の独自記法を用いてドキュメント中に記述することで表示させることができる .

4.2.1 表データの表示

通常の Wiki ページを編集する際、表データを挿入したい箇所に「&table(テーブル名);」と入力することで、指定の表データが表示される (図 6) . 更に、この記法に以下の引数を加えることで、様々な表データ操作を行うことができる .

特定列のみの表示 &table(テーブル名){|フィールド名 1|フィールド名 2|};

この場合は、フィールド 1 とフィールド 2 の列のみの表データが表示される (図 7) . この引数を指定しない場合は、全ての列を表示する .

行の昇順ソート &table(テーブル名){sort:フィールド名};

指定したフィールドをキーにして、行を昇順ソートした表データが表示される . 指定

したフィールド内のデータが全て数値だった場合は数値順ソート、それ以外の文字列が入っている場合は辞書順ソートとなる .

行の降順ソート &table(テーブル名){reverse:フィールド名};

行を降順ソートした表データが表示される . ソート順が変化する以外は昇順ソートの場合と同じである .

行ソート引数の優先順位

複数の sort 引数を用いることにより、ソートの優先順位を設定することが可能である (図 8) . 先に記述したキーが優先され、記述された順に優先順位が決定し、その状態でソートされ、表示される .

表の結合 &table(テーブル 1 名)(テーブル 2 名){join:フィールド名};

指定したフィールドのセル内容を元にテーブル 1 とテーブル 2 を結合した表データが表示される (図 9) .

複数の引数を設定

以上の引数を複数指定した場合は、記述した順番に操作が処理され、それによって構成された表データが表示される (図 9) .

4.2.2 SQL 文との比較

図 9 の操作をする場合を例に挙げる .



図 8 フィールド「EX」の昇順ソート、フィールド「bpm」の降順ソートの優先順位でソートを実行し、表示した。

SQL 文

```
SELECT ジャンル,EX,NoteEX
FROM popn17b,popn17a
WHERE popn17b.ジャンル=popn17a.ジャンル
```

独自の Wiki 記法

```
&table(popn17b)(popn17a){join:ジャンル}{|ジャンル|EX|NoteEX|};
```

SQL 文に比べて、括弧を用いて操作を区切っているため、加えられる操作が分かりやすく、実際に表示される表データをイメージしやすい。また、“|”を用いるなど、従来の Wiki 記法を意識した文法であるため、Wiki の編集に慣れたユーザであれば覚えやすく、有用な記法となっている。

5. 利用 例

現段階で考えられる、本研究の具体的な利用例を提案する。

5.1 Wiki システムとしての使用法

膨大な表データを扱うページを、Wiki システムを用いて構築する際に用いることができる。表データ内の内容の書き換えや行の追加が頻繁に行われた結果、非常に大きな構造のテーブルになった場合に有効であると考えられる。



図 9 二つのテーブル「pop17b」「popn17a」を、フィールド「ジャンル」をキーにして結合、フィールド「ジャンル」「EX」「NoteEX」のみを表示させた。

多量の情報を整理する CMS としての Wiki システム

本システムの活用例として、多量の情報が既に存在し、更に情報が追加されるような、まとめサイトを挙げる(図 10)。個々のユーザの意見や実際の事例も、他ユーザにとっては重要な情報となり得るため、結果として、多種多様な情報が一箇所に集まることになる。

それらの情報を管理する手段として、本研究で実装したシステムを用いることができる。例として、大きな表データをいくつか作成しておき、用途に応じて加工、操作を行った上でその表データを表示させる。これにより、閲覧者は求める情報を即座に発見することができる。また、表データは複数の場所で共有でき、一つの表データ編集を完了させるだけでその表データを用いているページ全体に適用される。そのため、作業の手間が大幅に軽減できる。

5.2 RDB インタフェースとしての使用法

今までの RDB ではデータ閲覧者に提示する際、用途に合わせてシステムを構築する必要があった。本研究では Wiki システムを用いており、データの提示を Wiki 記法を用いて容易に行うことができる。新規システムの構築や SQL 文の学習はいらず、必要な編集スキルは少量で済むため、幅広いユーザが RDB を用いたデータ管理を行うことが可能になる。複数ユーザが使用できるスケジューラ



図 10 例として、ゲームの情報ページの一つを作成した。



図 11 例として、2008 年 9 月の研究室内の統合スケジュールを作成した。

具体例として複数ユーザが管理するスケジューラを提案する(図 11)。まず、個人が日付とスケジュールを記入した表データを Wiki 記法を用いて作成する。そして、各々のスケジュールを日付をキーにして結合することで、そのコミュニティ内でのスケジュールを表示することができる。さらに、表の前後にアノテーションを挿入できるため、ユーザごとに活用方法を細かく変えることができ、自由度の高いスケジューラとして機能する。

6. まとめ

6.1 評価

多量の情報を整理できるシステムでは、従来の表データを用いて情報をまとめたサイトと比べ、編集が容易であり、効率が良く感じた。特に、複数ページに渡って表データの同じ内容が必要になることが多数あり、表データを共有し、操作を行うことでそのページに即した表データを表示できることは想像以上に便利であった。

スケジューラに関しては、残念ながら複数ユーザでの使用は行っていない。しかし、実際にデータからページを構築する際、ほんの一行の表データ表示用の独自記法を記述するだけで、閲覧しやすい統合スケジューラが表示でき、有用性のある使用方法であることが期待で

きた。

ところが、どちらの利用例に対しても、表データ自体の編集に Wiki 記法を用いている点に関して問題点が浮上した。実際に、表データが非常に大きくなった際、編集するドキュメント量の肥大化は無視できず、その結果、編集箇所を探す手間が発生したり、編集ミスを誘発してしまうことがあった。

6.2 結論

本研究では、まず現在の情報爆発を背景とした情報管理インタフェース、特に表データ管理の必要性を説明した。その上で、Wiki と RDB をキーワードに表データ編集における煩雑性と、必要となるユーザスキルの高さをそれぞれの問題点として提起した。その問題点を解決する手段として、RDB によって表データ操作を強化した Wiki システムの実装を行った。

また、現時点での利用例を挙げ、実際に情報を入力し、表データの構築を行った。その結果、編集の手間は軽減され、表データ操作を容易に行うことができ、提起した問題に対して有効な解決手段であることが示された。

しかし、本研究の中で、Wiki 記法自体が編集の煩雑性の一因となっている可能性が浮上

した。この点に関する解決策も今後模索していくと共に、Wiki システムの高い自由度を考慮し、他にも様々な用途を考える必要がある。新たに評価を行い、新機能の実装を進めていくことによって、問題点を解決することができれば、非常に有用なシステムになることが期待できる。

6.3 今後の課題

6.3.1 特定行のみの表示

特定の単語を指定すると、セル内にその単語が存在する行のみを表示する機能を提案する。大量の情報を整理するページにおいて表データを表示させる場合は、そのページに必要な行のみを表示させ、関係のない行を非表示にする。このような操作を行うことにより、閲覧者が求めている情報を的確に提示することができる。

一例として、表データを作成する際に分類用のフィールドを作っておく。表示時はそのフィールドのセル内の単語によって表示する行を選択し、更にフィールド自体を非表示にすることで、必要十分な情報のみが表示されたスマートな表データが構成される。以上のような、更に柔軟な操作が可能になると予想される。

6.3.2 表データの作成、編集方法の変更

現段階では、表データの作成や編集は全て従来の Wiki と同じ記法で行っている。しかし、結論で述べたように、表データ単体が非常に大きくなった場合、Wiki 記法では編集に手間がかかってしまうだけでなく、編集ミスも起きやすくなる。

そこで、セルの内容を編集する際には、Wiki 記法を用いずに、直感的に文字列を変更できるシステムを考えた。用意されたリンクをクリックすると、別ページに遷移することなく、行内やセル内のデータ編集、行の追加を容易に行うことができる。これにより、編集を直感的に、より簡単に行うことができる。

6.3.3 タグ機能

本システムでは非常に大量の表データが作成されることが予想され、そういった場合には、表データの内容だけでなく、表データの配置や分類なども考慮する必要が出てくる。

そこで、ソーシャルブックマークサイト¹³⁾におけるソーシャルタギングのような、システム内の情報を組織化、分類することができるタグ機能を考えた。一つの表データに複数のタグを指定することで、その表データの内容を端的に表したり、分類を行うことができる。タグ機能を用いることで、特定のタグが付けられた表データのリスト表示、タグの検索による必要な表データの探索など、分類や整理を行う場合に有用であると考えている。

参考文献

- 1) Bo Leuf and Ward Cunningham. *The Wiki Way: Quick Collaboration on the Web*. Addison-Wesley, 2001.
- 2) Roberto Tazzoli, Paolo Castagna, and Stefano Emilio Campanini. Towards a Semantic WikiWiki-Web *Poster Track, 3rd international Semantic Web Conference* pp.7-11, 2004.
- 3) Kouichirou Eto, Satoru Takabayashi, Toshiyuki Masui. qwikWeb: integrating mailing list and WikiWikiWeb for group communication. *Proceedings of the 2005 International Symposium on Wikis* pp.17-23, 2005.
- 4) D. Andresen, et al. The WWW Prototype of the Alexandria Digital Library, *Proceedings of ISDL '95* pp.17-27, 1995.
- 5) 畑田稔, 遠藤裕英. WWW-RDB 連携システムの開発. *Transactions of Information Processing Society of Japan* 38(2) pp.349-358, 1997.
- 6) 八木原勇太, 寺田実. Yagi Wiki: コンテンツの作成・整理が簡単に行える Wiki の作成. 第 49 回プログラミング・シンポジウム pp.41-48, 2008.
- 7) PukiWiki-official <http://pukiwiki.sourceforge.jp/>.
- 8) 自作プラグイン/areaedit.inc.php - PukiWiki-official <http://pukiwiki.sourceforge.jp/?自作プラグイン/areaedit.inc.php>.
- 9) 自作プラグイン/table.edit.inc.php - PukiWiki-official <http://pukiwiki.sourceforge.jp/?自作プラグイン/table.edit.inc.php>.
- 10) MySQL :: The world's most popular open source database <http://www.mysql.com/>.
- 11) phpMyAdmin <http://www.phpmyadmin.net>.
- 12) Wikipedia <http://ja.wikipedia.org/wiki/>.
- 13) Delicious - social bookmarking <http://delicious.com/>.