

## 並べ替えの手間

文章やメモなどを新しく作成するとき、情報を分類しながら追加したり、編集することがよくあります。アイデアやメモをまとめるときには、新しい情報は分類しながら追加していくとうまく扱えますし、役に立つ Web ページのリンク集を作るような場合も、内容別に分類しながらリンクを追加していくことが多いでしょう。買い物リストや TODO リストなどのちょっとしたメモはもちろんですが、プレゼンテーションの資料や原稿などの比較的長いものも、たいいていは内容をすこしずつ考えながら書き足していくのではないのでしょうか。

そのような場合、編集する情報の量が少ないうちは並べ替えもたいして苦にならなないので、普通のテキストエディタを利用すればよいでしょう。しかし、データの量が多くなってくると、並べ替えや編集に手間がかかるようになり、単純なテキストエディタでは効率的な編集ができなくなります。

編集作業では、関連する情報を近くに置いたり、なんらかの順番でソートしたり、あるいは分類の階層構造を修正したりといった操作が必要になります。しかし、大きな文書を扱う場合、単純なテキストエディタでは効率が極端に落ちてしまいます。たとえば、2つの大きなブロックの順番を入れ替えるとしましょう。そのようなときは、一方のブロックの先頭と末尾の位置を指定してコピーし、その後にもう一方のブロックの場所に移動して貼り付けたりします。この方式では、コピーする領域や貼り付ける場所を間違えたり、貼り付け操作そのものを失敗する可能性があるの

で、手軽にというわけにはいきません。

テキストの全体像を把握しながら移動などの操作をおこなうには、全体を縮小表示したり、なんらかのフィルタで内容を絞り込んだり、データを階層的に管理するといった工夫が必要です。そんなとき、階層的な情報を並べ替えながら、入力や編集を簡単におこなえるツールがあれば便利そうです。

大規模な情報を管理する場合、階層的な構造を利用するのが普通ですし、計算機上でこの種の情報を整理するツールもいろいろとあります。UNIX や Windows のファイルシステムは階層構造ですから、ファイル管理ツールで階層型データが扱えます。2005年3月号で紹介したような方法を使えば、これらのツールをデータ管理に応用することも可能でしょう。メニューや設定ファイルなども、階層的データ構造を前提としたつくりになっています。Windows では TreeView という GUI 部品で階層情報を管理することが多いようですが、その他の環境でも同様の方法で情報管理ができます。

このように、ファイルやメニューの管理については、階層的データ構造を前提としたツールがひろく使われています。ところが、通常メモやテキストの編集では、データの追加や階層的並べ替えを補助する方法はあまり利用されていないようです。たいかに、テキストを階層的に扱う“アウトライン・エディタ”があり、Emacs や MS Word にも“アウトライン・モード”があります。しかし、これらを十分に活用している人はあまりみかけません。たいいていは、コピーや貼付けなどの一般的な編集コマンドで間に合わせているのが実情のようです。

Emacs のアウトライン・モードの機能は、テキストを

階層化し、それらのレベルの表示/非表示を切り替えるのが基本です。しかし、階層のレベルの変更や、ブロックの並べ替えがとくに簡単にできるわけではありません。このあたりが、活用している人が少ない理由かもしれません。

ちょっとしたメモやテキストの作成や編集でも、分類や並べ替え操作は頻繁におこなわれます。ですから、簡単な操作でテキストの追加や階層化、並べ替えができるシステムがあれば便利でしょう。メモ、名簿、予定表、TODO リスト、ブックマークなどでは、追加や編集、並べ替えが頻繁におこなわれるので、このようなシステムは有効だと思います。もちろん、大規模なデータの場合も、階層的な管理や並べ替え操作が簡単にできれば扱いが楽になります。

こういったことを考えると、規模の大小にかかわらず、手軽に使えて有効な編集/並べ替えの技術が求められているように思います。

## 並べるシステム

そこで、ブラウザ上で手軽にデータを編集するために、Wiki で使える入力/編集システムを試作してみました。

私は、TODO リストや予定表、ブックマークなどをすべて 2001 年 11 月号で紹介した Wiki システムで管理しています。そこでまず、その上でデータの追加や編集を簡単におこなう方法を試してみました。これは、以下のような機能の実現を目指したものです。

- 単純な操作でデータの追加や並べ替えができる  
矢印キーを使い、選択したテキストの位置や階層を移動したり、表示する階層レベルを変化させたりします。
- 編集モードや書込みボタンを不要にする  
Wiki システムの大きな欠点は、編集モードと閲覧モードが分かれていることにあります。そのため、編集に手間がかかり、編集結果も書込みボタンを押さなければ反映されません。  
NOTA<sup>1</sup>などの先進的な Wiki では WYSIWYG 編集や自動書込みも可能ですが、テキストベースの従来型 Wiki の大半は、編集モードと書込みボタンをベースにしています。今回のシステムでは、編集をできるだけ簡略化しつつ自動書込みをおこなうことにします。

1 <http://rakusai.org/nota/>

図 1 Wiki ページ画面



### ●ズームングとフィルタリングを活用する

以前に、大規模なデータを効率よく検索するためのシステムとして LensBar (2000 年 2 月号) や Q-Pocket (2000 年 5 月号) を紹介しましたが、Wiki ページ上でもこれらの手法が使えるようにします。

## 動作例

Wiki 掲示板システムを JavaScript で拡張し、上記のような特徴をもつシステムを作ってみました。このシステムの概要を紹介します。

### 簡単な編集と保存

図 1 は、“並べる!技術”についての情報を Wiki ページ上で閲覧しているところです。

通常の Wiki ページでは、[編集] ボタンを押して図 2 のような編集モードに移行し、テキストの編集を終えたら、[Write] ボタンを押して書込みをおこないます。しかし、ちょっとした編集のために、これらのボタンをいちいち操作するのは面倒です。

今回のシステムでは、ボタン操作で編集モードに移行する代わりに、テキストをクリックするとその行が編集可能になるようにしてみました。たとえば、図 2 で“既存システムの問題点”の行をクリックすると、画面が図 3 のように変化してこの行が編集可能になります。

ここで、“大きな”という文字列を追加すると、画面は図 4 のように変わります。モノクロなので分かりにくいかもしれませんが、テキストの編集中は文字入力枠の背景色が

図 2 Wiki ページ編集画面



図 3 Wiki ページ編集画面



図 4 テキスト編集集中



変化します。編集結果は、2005年5月号で紹介した Ajax の手法を用いて随時サーバーに送られ、保存されます。保存されると、文字入力枠の背景色が通常の白に戻ります。

テキスト以外の場所をクリックすると文字入力枠が消え、編集後の Wiki ページが表示されます(図 5)。

図 5 編集後の Wiki ページ



図 6 項目の移動



このように、該当箇所の文字列をクリックするだけでテキストが編集でき、しかも自動的に保存されるため、Wiki ページ上での編集が手軽におこなえるようになりました。

### 項目の移動

選択した行とその下の階層の行とのあいだは、Shift キーを押しながら上下の矢印キーを押すことで移動できます。たとえば、図 5 の状態で「既存システムの大きな問題点」をクリックし、続けて Shift キーを押しながら下向き矢印キーを押すと、この階層以下に含まれる行がまとめて下に移動します(図 6)。

### 項目の追加

Shift キーを押しながら行をクリックするか、編集中に Enter キーを押すと、図 7 のように新しい行(項目)が追加されます。

図 7 新しい行の追加



図 8 項目の畳み込み



### 項目の畳み込み表示

左右の矢印キーを押すと、階層表示のレベルを切り替えることができます。たとえば、図 6 の状態で左向き矢印キーを 2 回押すと、下の階層が隠れて、一番上の階層だけが表示されます(図 8)。

### 項目のフィルタリング

Search の欄に検索文字列を入力すると、Migemo<sup>2</sup>を用いたテキストのフィルタリングがおこなわれます。例に使っているテキストに対して“sou”と入力した場合、sou(そう)という読みをもつ単語(この例では“操作”)を含む行だけが表示されます(図 9)。

ここで左右の矢印キーを押すと、LensBar と同じ要領で、検索文字列にマッチした行と、それよりも上のレベルの階層の行も表示できるようになります。

<sup>2</sup> <http://namazu.org/~satoru/migemo/>

図 9 インクリメンタルなフィルタリング



図 10 表示される階層レベルの制御



今度は、図 9 の状態で右矢印キーを押してみましょう。すると、一番上の階層の行も一緒に表示されます(図 10)。

このような手法は“Focus+Context”と呼ばれ、注目している行と文書全体の構造を同時に眺められる点に特徴があります。

### 評価

この手法を導入した Wiki ページをしばらく使っていますが、たいへん快適です。従来は、Wiki にデータを入力したあとで並べ替えるのが面倒だったため、よほどのことがなければ入力したままにしていました。しかし、このシステムの場合は、入力したデータの編集や並べ替えが簡単なので、Wiki 上での編集作業を負担に感じることもありません。これまで、Wiki データを編集するときは編集モードに切り替えていましたが、並べ替えや編集が直接的に

実行できるのはたいへん便利です(最近、なぜデータをわざわざ手作業で保存していたのだろうと不思議に感じるほどです)

現在の実装では、編集をおこなうたびに新しい Wiki ページが生成されてしまいます。これが気になる人もいると思いますが、そもそもテキストデータが膨大なサイズになることはあまりありませんし、なによりも編集時のログがすべて残るという安心感があります。実装方法に関しては、まだまだ考慮の余地が残っているものの、大まかな方針としては今回の手法はかなり正しい方向にあるように思います。

---

## おわりに

今回の方法は、単純なテキストデータの編集を前提としています。しかし、テキストは方法によってはいろいろに解釈できるので、実質的に制限はほとんどないといってもよいでしょう。

私の Wiki ページでは、予定表をテキストデータとして編集し、それをフィルタにかけてカレンダー形式で表示するようにしています。このように、適当なフィルタに通したあとで閲覧する方式であれば、直接には単純なテキストしか編集できなくても、多様な見せ方が可能になります。

今回の方法を使えば、大規模な YAML<sup>3</sup>などのデータも扱えるはずです。YAML のデータも簡単な形式のテキストですが、XML と同様に複雑なデータ構造が表現できます。XML データを手作業で編集するの大変ですが、今回のようなシステムと YAML を併用すれば、手作業によるデータの作成や処理も難しくないでしょう。

計算機上でおこなうさまざまな仕事のなかでも、データの入力や編集はもっとも基本的で重要な機能です。このシステムを足掛かりとして、さらに便利な入力/編集手法を開発しようと考えています。

(ますい・としゆき 産業技術総合研究所)

---

<sup>3</sup> <http://www.yaml.org/>