

インターフェイスの街角 – Web で Office

増井 俊之

ひと昔ほど前、Web ブラウザさえあれば、計算機のデスクトップ環境や MS Office のような事務用ソフトウェアは不要になるだろうという予測がよく話題になっていました。しかし、現在の一般的な計算機の利用状況をみると、どうやら Web のブラウジングと事務処理は別物と考えられているらしく、両者の融合はそれほど進んでいないようです。

Web と完全に融合したデスクトップのオフィス・アプリケーションが登場するまでには、まだしばらく時間がかかりそうです。ただし、ある程度の事務処理であれば、現時点で簡単に手に入るブラウザや CGI プログラムを用いておこなうことができます。今回は、その方法を紹介しません。

Wiki による事務処理

Web 上での情報整理システム Wiki Wiki Web(以下、Wiki と略)を 2001 年 4 月号で、Wiki と掲示板を統合したシステムを 2001 年 12 月号で紹介しました。最近、私はこの Wiki 掲示板システムをあらゆる作業に利用しようと試みています。

オリジナルの Wiki は、Web 上での円滑な情報共有を第一目的として開発されたためか、体裁をきれいに整えたり、四則演算などをおこなう機能はなく、リンクと最小限の文書整形の機能しかありませんでした。

Wiki が普及するにつれ、これをさらに強力にしようとする人たちが数多く現れ、現在ではかなりの数の Wiki クローンが公開されています。昨年 12 月号で紹介した Wiki 掲示板もその 1 つです。これらの Wiki クローンでは、オリジナルの Wiki に対して以下のような拡張が加

図 1 Wiki データの編集

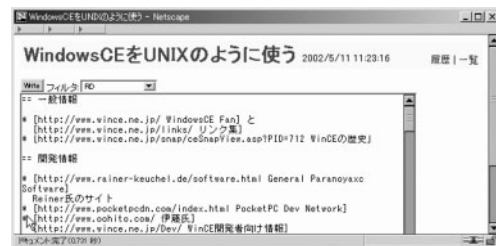
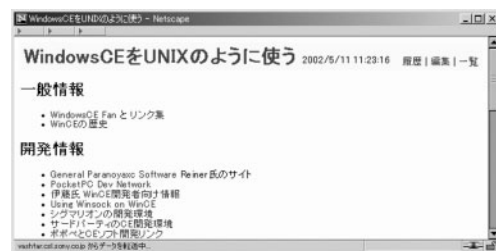


図 2 RD による整形結果



えられています。

- 超整理法的機能
- 履歴情報の追加
- CGI による計算機能
- 整形タグの拡張

一般に、Wiki ではユーザーが単純なテキストデータを編集し、その結果を書き込んで Web ページを生成するときにデータを整形する仕組みになっています。たとえば、図 1 の例では、情報を RD 形式¹で記述し、これを書き込むと HTML タグが追加されて、図 2 のようなページが生成されます。

¹ Ruby のプログラムにドキュメントを埋め込むための形式です。詳細は、<http://helium.ruby-lang.org/~rubikitch/RDP.cgi?cmd=view&name=RD> を参照してください。

この例では RD を用いてテキストから HTML に変換していますが、テキストデータを適切な形式に変換するフィルタを用意すれば、通常の事務処理や PDA 上での作業の大半を Web 上でおこなうことができます。これを実現するために、Wiki 掲示板システムにフィルタを追加し、各種の事務処理に応用する実験を進めています。

デスクトップ計算機や携帯端末をビジネス用途で使う場合、ワードプロセッサや表計算、プレゼンテーション、メールなどのプログラムがよく利用されます。また、携帯端末では基本的機能として、住所録や予定表、TODO 管理、辞書、メモ帳などが用意されています。これらを Wiki でうまく扱うことができれば、Web の閲覧だけでなく、ほとんどの事務処理を Wiki 上でおこなえるようになるでしょう。

Wiki 上での事務処理には、次のような利点があります。

- 共同作業を円滑に進められる。
- どこからでも同じデータにアクセスできるので、データを同期させる必要がない。
- ワードプロセッサ、メール、予定表、プレゼンテーションなどのデータに対し、同一の方法で検索を実行できる
- フォルダやファイル名を気にすることなく、作成時刻やリンクなどで管理できる。

以下では、各種の事務アプリケーションと同等の作業を Wiki でおこなう試みについて、いくつかの例を紹介します。

トップページ

Wiki ページの参照および検索は、すべて図 3 のようなトップページからおこないます。日付と項目のあいだにある `` ` `` や `` ` `` などは、過去にどのくらいそのページにアクセスしたかを簡単に視覚化したものです。黒色に近いほどアクセスが多いことを表し、左側が最近の状況、右側が以前の状況を示しています。

ワードプロセッサ

オリジナルの Wiki には、表 1 のタグを用いた簡単な整形機能があり、多くの Wiki クローンでもこれらが継承されています。

これらのタグだけで凝った体裁にするのは不可能なので、Wiki ページの外見はどれも似たような感じになりが

図 3 Wiki ページのリスト



表 1 Wiki Wiki Web の整形規則

タグ	意味
WikiName	別ページへのリンク
空行	<p>
----	<hr>
Tab *	リスト項目レベル 1 ()
Tab Tab *	リスト項目レベル 2
Tab 1	番号付きリスト項目 ()
Tab 項目: 説明	定義リスト項目 (<dt/><dd>)
行頭空白文字	<pre> 指定
Tab 空白文字 : Tab	引用
'' 文字列 ''	斜体表示 (<i> ~ </i>)
''' 文字列 '''	強調表示 (~)
URL	リンク

ちです。しかし、HTML のタグを駆使すれば、一般の Web ページと同程度に体裁を整えることができます。

直接 HTML を記述するには慣れが必要で、入り組んだ表などになると熟練者でもけっこう苦労します。しかし、Netscape に付属している GUI の HTML エディタ Composer を使うと、複雑な段組なども比較的簡単に記述できます。

図 4 は、Netscape Composer で履歴書を作成しているところです。罫線を多用するために入れ子のテーブルを利用しています。

このようにして作成した HTML テキストを、今度は Wiki のデータとして編集します(図 5)そして、[Write] ボタンを押してこれを保存すると、図 6 のような Web ページが表示されます。

Wiki だけですべてを処理しようとするの大仕事になってしまいますが、発想を転換して Composer などの HTML 編集ツールで骨格を作り、Wiki ページとして保存すれば、複雑な体裁の文書も容易に作成できます。

図 4 Composer で編集



図 5 Wiki で履歴書 HTML を編集



図 6 作成された履歴書 Wiki ページ



プレゼンテーション

Netscape 7²には“全画面表示モード”があり、画面上からメニューなどの GUI 部品を完全に隠せるように

2 Netscape 7は、Netscape 6に比べて、タブが使える、全画面表示が可能といった特徴があります。執筆時点では Windows 用の Preview Release しか公開されていませんが、Mac OS 版、UNIX 版なども近々公開されるでしょう。

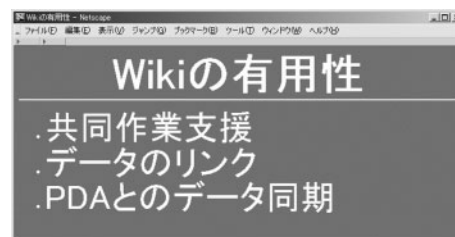
図 7 プレゼンテーション資料のテキスト



図 8 プレゼンテーションの目次



図 9 プレゼンテーションのページ



なっています³。この機能を利用すると、PowerPoint や MagicPoint などのツールの代わりに Web ブラウザを使ってプレゼンテーションをおこなうことができます。

MagicPoint の記述ファイルと似た形式のテキストファイルを Wiki データとして作成し、書き込む際にスライドページに変換するフィルタを使えば、HTML でもかなりきれいなプレゼンテーション・ファイルが作れます。

図 7 は、プレゼンテーション資料のテキストを作成しているところです。字下げでページと項目の深さを表現しており、字下げのない行がスライドのタイトルになります。

このテキストを保存するときにプレゼンテーション用のフィルタを指定すると、図 8 の目次が Wiki ページとして生成されます。各ページは、図 9 のように表示されます。

このスライドはテキストだけで構成されていますが、

3 Internet Explorer にも全画面モードはありますが、ページ移動時に一瞬メニューが表示されるため、プレゼンテーションには不向きです。

図 10 キー入力を解釈する JavaScript プログラム

```
document.onkeypress = keypress;
function keypress (event)
{
  var i = keycode(event);
  if(i == 0x20){
    window.location.href = "foobar.html";
  }
}
function keycode(event)
{
  if(navigator.appName.indexOf("Microsoft") =>
    != -1)
    return window.event.keyCode;
  if(navigator.appName.indexOf("Netscape") =>
    != -1)
    return event.which;
}
(誌面の都合上、=>で折り返しています)
```

 タグで画像を入れたり、MING⁴を用いた Flash ムービーを使えば、PowerPoint や MagicPoint と遜色のないプレゼンテーション資料を作成することも可能です。

さらに、図 10 のような JavaScript プログラムを利用すると、全画面モードでもキー入力によるページ移動が可能になります。

表計算

計算を実行するフィルタを用意すれば、通常は表計算アプリケーションを使うような計算も、ある程度は Wiki で処理できます。

PHP や eRuby (embedded Ruby)⁵などが使える環境なら、計算式をそのまま HTML に埋め込み、簡単な計算を実行させることができます。

テキストエディタ上で表計算をおこなうシステムの 1 つに、Mule 上で使える Boomborg Keisan⁶があります。このシステムでは、文字列で表現した式などを評価して置き換え、表計算や繰り返し計算などを簡単に実行させることができます。

図 11 は、Boomborg Keisan で所得税の計算をしているところです(Boomborg Keisan のサンプルファイルを借用しています) たんなるテキストに見えます

4 <http://opaque.net/ming/>
 5 <http://www.modruby.net/>
 6 <http://nicosia.is.s.u-tokyo.ac.jp/boomborg/boomborg-keisan-j.html>

図 11 Boomborg Keisan による計算の例



図 12 会議室予約の入力



が、じつは給与所得の“7020000”という数字はその上の“700000”と“20000”を足して自動計算されたものです。

Boomborg Keisan では、テキスト中に “ + ” という記号があると、その下の数字をすべて加算し、“-----” がみつければ、その下に合計を記入するという規則が定義されています。これによって、一般の表計算システムと同様に、自動的に計算させることができるわけです。計算結果などにラベルを付けて別の場所で参照したり、繰り返し処理を指示することも可能です。

Boomborg Keisan システムは、Mule でちょっとした計算をするときに便利です。これと同様の機能をもつフィルタを用意しておけば、Wiki ページを書き込むたびに計算処理をおこない、その結果を表示したり、新しいページに反映させることができます。

予定表

日付データをテーブルに変換するフィルタを使うことにより、グループウェアの機能の定番ともいべき会議室予約システムなどが簡単に作れます。

予定表の内容は図 12 のようにテキスト形式で編集し、フィルタを通して表の体裁に変換すれば、図 13 のようなカレンダーとして閲覧できます。

図 13 会議室予約ページ



データベース

PDA では、住所録や店の情報などを記録しておくために、簡易型データベースがよく使われています。同じ形式のデータの集合を適当に整形して表示させるフィルタを用意すれば、閲覧しやすくなります。

図 14 は、Web ページと連動させた単語帳を作成しているところですが（英単語の使用例には、Web 上のテキストを使っています）このソーステキストに対して、英単語と日本語訳、例文を整形するフィルタを適用すると、図 15 のような見映えのいい単語帳が作れます。

おわりに

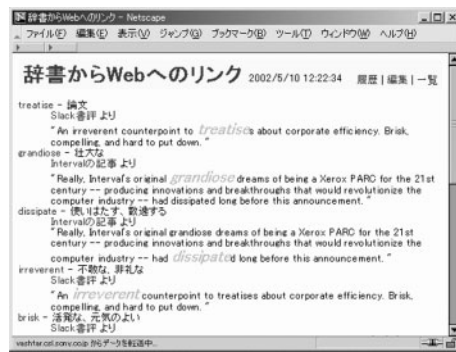
携帯端末や携帯電話の普及にともない、これらの機器でのデータ管理や機器間でのデータ同期の手法の確立が大きな課題となりつつあります。Palm の HotSync をはじめ、データ同期を簡単におこなうための枠組みがいろいろと提案されていますが、これらの機構はユーザーが明示的に実行することを前提としています。そのため、同期を忘れてしまったり、肝腎なときにかぎって手許の計算機に必要なデータがなかったりということがよく起きます。この問題は、一朝一夕に解決できるものではなさそうです。しかし、ブロードバンド環境がかなり普及してきたので、複数の機器のデータをつねに同期させたうえで PDA などを持ち歩く方法はすたれていくかもしれません。今後は、インターネット上に置いた 1 つのデータをいろいろな機器から参照したり、修正したりする方式が主流になるのではないでしょうか。その意味でも、あらゆるアプリケーションを Web 上で扱うことは大きな意味があると思います。

一般のアプリケーションと比較した場合、Wiki のインターフェイスは優れているとはいえませんし、できるこ

図 14 単語帳編集



図 15 単語帳ページ



とも限られています。WYSIWYG ではなく、つねにテキストとして編集しなければならないのも苦しいところです。とはいえ、テキストデータの活用という面では便利であり、データをネットワーク上に置く利点はかなり大きいので、現時点では Wiki の使用は有効な選択肢の 1 つだと思います。

将来的には、Microsoft の .NET をはじめとする各社のネットワーク事業の一環として、Web と連動した使いやすい事務処理アプリケーションが浸透していくでしょう。いまのところ、私は Web 上で原稿を書いているわけではなく、プログラムも Web と無関係な場所で開発しています。しかし、Web から TeX を動かしたり、HTML をより深く使いこなしていけば、文書の作成やプログラム開発も Web 上で違和感なくおこなえるようになるかもしれません。

(ますい・としゆき ソニー CSL)