

インターフェイスの街角 (25) – i モード 端末の活用

増井 俊之

最近、NTT DoCoMo の i モード 端末の売行きが好調なようです。これほどヒットした要因の 1 つは、新聞社や金融機関などによる i モード 専用のサービスが充実している点にあるといわれていますが、携帯電話ひとつで電子メールの交換や Web ページの閲覧ができる点も人気を集めている理由の 1 つでしょう。今回は、`Compact HTML` で記述した Web ページを使って i モード 端末を活用する方法を紹介します。

Compact HTML

i モード 端末では、Compact HTML で記述された Web ページが表示できます。Compact HTML は携帯端末向けの HTML 4.0 のサブセットで、1998 年にアクセス社¹を中心とする国内の数社により W3C に提案されました [1, 2]。

携帯電話上で情報を提供するためのプロトコルとしては Motorola や Nokia などの電話会社が WAP (Wireless Application Protocol)² を推進していますが、Compact HTML の強みは既存の Web ページの資産を活用できる点にあります。

タグ

携帯端末では、背景色やフォントの大きさ、色などを小刻みに変えたり、細かな表示をするのは困難です。そこで、Compact HTML では `` や `<table>` などは省略しています。しかし、基本的な整形用のタグ (`
` や ``、`<center>` など) やリンク用のタグ (`<a>` など) 画

表 1 i モード 端末で使えるおもなタグ

<code><A></code>
<code><BLOCKQUOTE></code>
<code>
</code>
<code><CENTER></code>
<code><DL></code> <code><DT></code> <code><DD></code>
<code><DIR></code> <code><MENU></code> <code></code> <code></code> <code></code>
<code><DIV></code>
<code><HR></code>
<code></code>
<code><INPUT></code>
<code><P></code>
<code><PRE></code>
<code><SELECT></code> <code><OPTION></code>
<code><TEXTAREA></code>

<http://www.nttdocomo.co.jp/i/tag/>

像表示用のタグ (`` など) ユーザーからの入力支援する `<input>` タグといった基本的なものに関しては通常の HTML と同様に利用できます。

i モード 端末では、Compact HTML のサブセットが使えます (表 1)

ダイレクトキー機能

Compact HTML は、携帯電話などのテンキーをもつ機器で効果的なブラウジングをおこなうために、HTML 4.0 で定義されている `アクセスキー (ダイレクトキー) 機能` を備えています。通常、HTML の `<a>` タグでは、

```
<a href="http://www.foo.com/">Foo</a>
```

のような記述で `www.foo.com` へのリンクを表現し、文字列 `Foo` をクリックして選択することによって指定された URL へ移動できます。

一方、Compact HTML では、

¹ <http://www.access.co.jp/>

² <http://www.wapforum.org/>、<http://www.dion.ne.jp/wap/>

```
<a href="http://www.foo.com/" accesskey=1>
Foo</a>
```

のように `accesskey` 属性で指定したキーを用いて指定 URL に直接移動するようになっています。上の例では、`1` のキーを押すと `www.foo.com` が選択され、そのページに移動します。以下のようにキーと URL の対応付けを明示すれば、`Foo` が `1` キーに対応していることをユーザーに示せます。

```
<a href="http://www.foo.com/" accesskey=1>
(1)Foo</a>
```

ダイレクトキー機能は `<a>` タグのほか、`<input>` タグのなかでも指定できます。たとえば、

```
<input type=submit accesskey=#>
```

と記述すれば、電話の `#` キーで入力フォームが送信できます。

電話番号の指定

i モード 端末では、`<a>` などのタグのなかで、URL やメールアドレスだけでなく電話番号を指定することもできます。たとえば、

```
<a href="tel:01-234-5678" accesskey=1> x社</a>
```

などと記述しておけば、これを選択した場合に該当する番号に電話が発信されます。

絵文字

i モード 端末では `` タグによる GIF 画像の表示も可能ですが、これには比較的時間がかるので、代わりに約 160 種類の絵文字(外字)が用意されています³。

i モード 端末の活用

i モード 端末では、Compact HTML で記述したインターネット上の任意の Web ページが閲覧できます。通常の CGI プログラムも使えますし、`<input>` タグを用いたユーザーの入力の送出手も可能なので、画面が小さいという制限はありますが、通常の Web ページと同等なインターフェイスを実現することができます

i モード 端末には、通常の PDA に備わっている電卓や住所録などの機能はありません。しかし、Web サーバー

³ <http://www.nttdocomo.co.jp/i/tag/emoji/>

図 1 週間予定表

長期 次週
11/14(日) 七五三
11/15(月) 13:00 部内ミーテ ィング
11/16(火)

図 2 長期予定表

11月
日 月火水 木金土
14 151617 181920
21 222324 252627
28 2930 1 2 3 4
12月
日 月火水 木金土
5 6 7 8 91011

上に住所録を置いて参照したり、あるいは CGI プログラムで計算をおこなうようにすれば、PDA と同様の処理が可能になります。

最近の携帯端末ではホスト計算機との同期が重視されています。これについても、つねにホスト計算機の情報だけを扱うのであれば、同期やデータ転送について気にする必要はなくなるので、場合によってはデータを個々の端末ごとにもっているより有利かもしれません。

予定表

1999 年 5 月号で、Web ブラウザを用いた個人情報管理の手法について解説しました。毎日の予定や日記を `1999/12/3/event` といった形式のファイルに格納し、これらをまとめて 1 つの大きな HTML ファイルにしておけば、スケジュールの参照や管理が手軽におこなえます。このようなデータを i モード 端末からも参照できるようにしておくと、どこにいてもスケジュールの確認ができて便利です。

5 月号では、`<table>` タグを用いて月間カレンダー形式のスケジュール表を作成しました。しかし、i モード 端末では `<table>` タグが使えませんし、画面のサイズも小さいのでこのままでは不便です。そこで、末尾のリスト 1 のようなプログラムを作成し、週間予定表と年間予定表とに分けてみました(図 1~2)。これならば、i モード 端末でも問題なく予定表を参照できます。

住所録

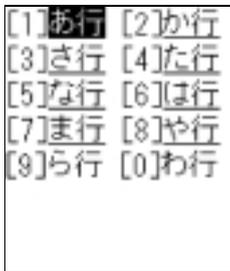
予定表と同様、住所録も i モード 端末から参照できるようにしておくと便利です。

画面が小さいので、リスト 2 のような HTML ファイルを使って名前を階層的に検索するようになります。図 3 は、この HTML ファイルを i モード 端末で表

リスト 2 階層的な検索を可能にする

```
<html>
<head>
<title>住所録</title>
</head>
<body>
[1]<a href="as.html" accesskey=1>あ行</a>
[2]<a href="kas.html" accesskey=2>か行</a><br>
[3]<a href="sas.html" accesskey=3>さ行</a>
[4]<a href="tas.html" accesskey=4>た行</a><br>
[5]<a href="nas.html" accesskey=5>な行</a>
[6]<a href="has.html" accesskey=6>は行</a><br>
[7]<a href="mas.html" accesskey=7>ま行</a>
[8]<a href="yas.html" accesskey=8>や行</a><br>
[9]<a href="ras.html" accesskey=9>ら行</a>
[0]<a href="was.html" accesskey=0>わ行</a><br>
</body>
</html>
```

図 3 住所録を表示



示したところです。

CGI を利用した電卓

ダイレクトキー機能で CGI を起動すれば、i モード端末を電卓として利用することもできます。末尾のリスト 3 の CGI プログラム calc.cgi は、かつて Hewlett-Packard の電卓で採用されていた逆ポーランド法による電卓を i モード端末上で使えるようにしたものです。calc.cgi を呼び出すと、i モード端末には電卓画面(図 4)が表示されます。

数字キーが押されると、そのキーを引数としてふたたび calc.cgi が呼び出されます。画面は、たとえば 2 を押すと図 5 のようになり、さらに `` を選択すると、図 6 のように変化します。

キーを押すたびにサーバーと通信するのでひどく遅いの
が難点ですが、サーバーを介してどんな計算でも実行でき
ます。

図 4 電卓の初期画面

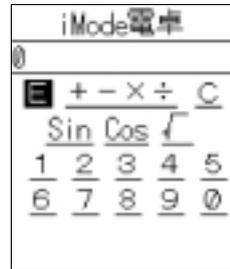


図 5 ``2``キーを押す

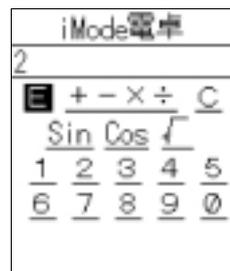
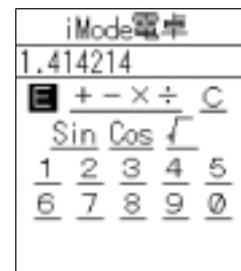


図 6 `` `` を選択



CGI を使った 5 択問題

暇つぶしに、i モード端末でクイズを楽しむというのは
どうでしょうか。MS-DOS の世界では、昔から ``5 択(ご
たく)`` というクイズプログラムの人気が高く、現在でも
Web 上でたくさんのクイズデータやプログラムが公開さ
れています⁴。5 択にはいろいろなバリエーションがある
ようですが、基本的には 1 つの設問に対して解答が 5 個
ずつ提示され、そのなかから正しいものを選ぶというゲー
ムです。たくさん問題が公開されていますし、問題も解
答も短いので i モード端末で遊ぶのにはちょうどいいの
ではないでしょうか。CGI プログラムと問題ファイルを用
意すれば、i モード端末で簡単に 5 択ゲームが楽しめます。

まず、図 7 のような問題のテキストファイルを用意し
ます。インデントされた 5 行が答の候補で、最初の答が正
解です。

5 択ゲーム用の問題は ``.5tq`` という拡張子のデータで、
Vector などの Web/FTP サイトで配布されています。
理由はよく分かりませんが、この 5tq ファイルは問題や
解答が独自のバイナリ形式で格納されています。そこで、
リスト 4 の Perl プログラムを使って図 7 のようなテキス

⁴ Vector のサイト (<http://www.vector.co.jp/vpack/filearea/win/game/quiz/5taku/>) では、たくさんの ``5 択`` 問題が公開されています。CGI や JavaScript による ``5 択`` のページもあります。

図 7 5 択問題の例

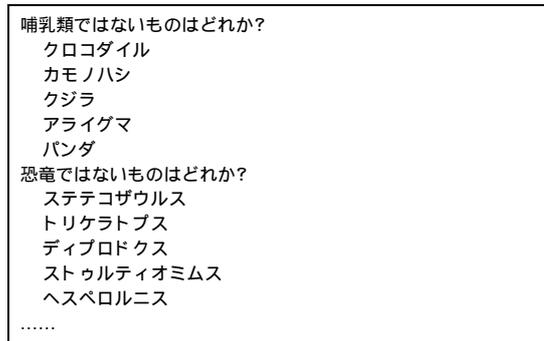
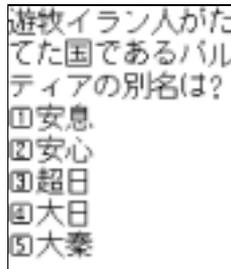


図 8 iモード端末での 5 択画面



ト形式に変換して使うことにします。

さらに、このデータに対応した CGI プログラム 5tq.cgi (リスト 5) を用意すれば、図 8 のような画面で 5 択問題を楽しむことができます。

この例では、i モード端末の絵文字を使って答の番号を表現しています。絵文字は、“&63879;”のように数字を使って指定することができます。

おわりに

今回は、Compact HTML で記述した Web ページを用いて i モード端末をさらに便利に使う方法を紹介しました。

CGI や入力機構を利用して柔軟なサービスを自力で作れるところが興味深いのではないのでしょうか。

i モード端末のように画面のごく小さい機器のことを、最近“Baby Face”と呼ぶのだそうです。携帯端末や Wearable Computing の計算能力はかなり高くなってきましたが、小さな画面をうまく使うためのインタラクション技術の研究はまだ不十分です。HTML は、もともと Baby Face 向けに設計されたものではありません。そのためか、Compact HTML も携帯端末向けの工夫がなされているとはいえ、i モード端末などの Baby Face 機器に最適とまではいえないように思います。将来的には、ズームングなどを使用した、新しい携帯端末向けのインターフェイスやプロトコルが望まれるところです。

(ますい・としゆき ソニー CSL)

[参考文献]

- [1] Tomihisa Kamada, *Compact HTML for Small Information Appliances*, February 9, 1998 (<http://www.w3.org/TR/1998/NOTE-compactHTML-19980209>)
- [2] Tomihisa Kamada and Joe Leung, *HTML for Information Appliances*, August 21, 1998 (<http://webreview.com/wr/pub/98/08/21/feature/>)

リスト 1 makeicalendar(予定表作成)

```
#!/usr/local/bin/perl

$nkf = "/usr/local/bin/nkf";
$rootdir = "/user/masui/DOC";          # 予定表を格納するディレクトリ
$scheduledir = "/www/masui/imode/schedule"; # iモード用の予定表ディレクトリ

$daysec = 24*60*60;
$weeksec = $daysec * 7;
$maxweek = 30;
@dayname = ('日','月','火','水','木','金','土');

sub getdate { # 時刻から年月日を得る
    local($t) = @_;
    local($sec,$min,$hour,$mday,$mon,$year,$yday,$isdst);
    ($sec,$min,$hour,$mday,$mon,$year,$yday,$isdst) =
        localtime($t);
    if($year >= 70 && $year < 100){ $year += 1900; }
    elsif($year < 40){ $year += 2000; }
    $mon++;
}
```

```

    ($year,$mon,$mday,$hour,$min,$sec);
}

# 週の最初の日の計算
$basetime = time;
($sec,$min,$hour,$mday,$mon,$year,$yday,$isdst) = localtime($basetime);
$basetime -= $daysec * $wday;

# 長期予定表のページを生成
open(list, "| $nkf -s > $scheduledir/index.html");
print list <<EOF;
<html>
<head><title>長期予定表</title></head>
<body>
<pre>
EOF

$prevmmon = -1;
for($week=0;$week<$maxweek;$week++){
    ($year,$mon,$mday) = &getdate($basetime + $week * $weeksec);
    if($week==0 || $mon != $prevmmon){
        print list "${mon}月\n";
        print list "日 月火水 木金土\n";
        $prevmmon = $mon;
    }
    print list "<a href=\"week$week.html\">";
    for($wday=0;$wday<7;$wday++){
        ($year,$mon,$mday) = &getdate($basetime +
            $week * $weeksec + $wday * $daysec);
        print list sprintf("%2d",$mday);
        if($wday==0 || $wday==3){ print list " "; }
    }
    print list "</a>\n";

    &doweeek($week);
}

print list <<EOF;
</pre>
</body>
</html>
EOF

close(list);

sub doweeek { # 指定された週の予定表ページを生成
    local($week) = @_;
    open(week, "| $nkf -s > $scheduledir/week$week.html");
    print week "<html><body>\n";
    print week "<a href=\"index.html\">長期</a> ";
    if($week > 0){
        $n = $week-1;
        print week "<a href=\"week$n.html\">前週</a> ";
    }
    if($week < $maxweek-1){
        $n = $week+1;
        print week "<a href=\"week$n.html\">次週</a> ";
    }
    print week "<hr>\n";
    for($wday=0;$wday<7;$wday++){
        ($year,$mon,$mday) = &getdate($basetime +

```

```

    $week * $weeksec + $daysec * $wday);
print week "$mon/$mday($dayname[$wday])<br>\n";
$file = "$rootdir/$year/$mon/$mday/event";
open(in,$file);
while(<in>){
    chop;
    print week "$_<br>\n";
}
print week "<hr>\n";
close(in);
}
print week "</body></html>\n";
close(week);
}

```

リスト 3 calc.cgi(電卓用 CGI)

```

#!/usr/local/bin/perl

# CGIの引数を取得
if($ENV{'REQUEST_METHOD'} eq 'GET'){
    $_ = $ENV{'QUERY_STRING'};
}
else {
    $len = $ENV{'CONTENT_LENGTH'};
    read(STDIN,$_,$len);
}
# 'arg=value'という形式のとき、$argにvalueを代入
s/\n//g;
@a = split(/&/,$_);
for $_ (@a){
    ($key,$val) = split(/=,$_);
    eval "\$$key = $val;";
}

sub enter { # スタックをpush
    $v3 = $v2; $v2 = $v1; $v1 = $v;
}

if($k ne ''){ # 数字キーの入力
    if($c){ &enter; $v = ""; }
    $v = ($v eq '0'? $k : $v.$k);
    $c = 0;
}
elsif($enter){ # HP電卓のEnterキー
    &enter;
    $c = 1;
}
elsif($clear){
    $v = '0';
    $c = 1;
}
elsif($comma){ # 小数点
    $v .= '.';
    $c = 0;
}
elsif($op){ # *, +, etc.
    $v = eval "$v1 $op $v";
    $v1 = $v2; $v2 = $v3;
    $c = 1;
}

elsif($fun){ # 数学関数の呼出し
    $v = eval "$fun($v)";
    $c = 1;
}
$v = '0' if $v eq '';
$d = ($v =~ /\./ && $c ? sprintf("%f",$v) : $v);

$s = "v=$v&v1=$v1&v2=$v2&v3=$v3&c=$c";

print <<EOF;
Content-type: text/html

<HTML><HEAD><TITLE>iMode電卓</TITLE></HEAD>
<BODY>
<center>iMode電卓</center>
<hr> $d <hr>
<center>
<a href="calc.cgi?enter=1&$s" accesskey=#>E </a>
<a href="calc.cgi?op=+&$s">+ </a>
<a href="calc.cgi?op=-&$s">- </a>
<a href="calc.cgi?op=*&$s">x </a>
<a href="calc.cgi?op=/&$s">÷ </a>
<a href="calc.cgi?clear=1&$s">C </a>
<br>
<a href="calc.cgi?fun=sin&$s">Sin</a>
<a href="calc.cgi?fun=cos&$s">Cos</a>
<a href="calc.cgi?fun=sqrt&$s"> </a>
<a href="calc.cgi?comma=1&$s" accesskey=#></a>
<br>
<a href="calc.cgi?k=1&$s" accesskey=1>1 </a>
<a href="calc.cgi?k=2&$s" accesskey=2>2 </a>
<a href="calc.cgi?k=3&$s" accesskey=3>3 </a>
<a href="calc.cgi?k=4&$s" accesskey=4>4 </a>
<a href="calc.cgi?k=5&$s" accesskey=5>5 </a>
<br>
<a href="calc.cgi?k=6&$s" accesskey=6>6 </a>
<a href="calc.cgi?k=7&$s" accesskey=7>7 </a>
<a href="calc.cgi?k=8&$s" accesskey=8>8 </a>
<a href="calc.cgi?k=9&$s" accesskey=9>9 </a>
<a href="calc.cgi?k=0&$s" accesskey=0>0 </a>
</center>

</BODY></HTML>
EOF

```

リスト 4 5tqdecode(5tq ファイルをテキスト形式に変換)

```
#!/usr/local/bin/perl

$file = shift;
open(in,$file) || die "Can't open $file";

# ヘッダ情報の読み飛ばし
read(in,$s,2048);

# 各問題の処理
while(read(in,$s,256)){
    $q = substr($s,0,116);
    $_ = &decode($q);
    s/ *$//;
    print "$_\n";
    for($i=0;$i<5;$i++){
        $a[$i] = substr($s,116+$i*28,28);
        $_ = &decode($a[$i]);
        s/ *$//;
        print "\t$_\n";
    }
}

# 問題/答のデコード
sub decode {
    local($s) = @_;
    local($c,$r,$o);
    $doit = 1;
    while($s =~ s/[\000-\377]//){
        $c = $&;
        if($doit){
            $o = ord($c) ^ 0x80;
            if($c eq ' '){ }
            elsif((( $o > 0x7f) && ( $o < 0xa0) ) ||
                (( $o > 0xdf) && ( $o <= 0xff))){
                $doit = 0;
                $r .= sprintf("%c",$o);
            }
            else {
                $r .= sprintf("%c",$o);
            }
        }
        else {
            $o = ord($c) ^ 0x80;
            $r .= sprintf("%c",$o);
            $doit = 1;
        }
    }
}
$r;
}
```

リスト 5 5tq.cgi(5 択実行 CGI)

```
#!/usr/local/bin/perl

# 5択データの読出し
$questions = 0;
open(in,"data");
while(<in>){
    chop;
    $q{$questions} = $_;
    for($i=0;$i<5;$i++){
        $_ = <in>;
        chop;
        $a{$questions,$i} = $_;
    }
    $questions++;
}
close(in);

# 乱数による問題選択
srand(time);
$q = int(rand($questions));
for($i=0;$i<5;$i++){
    $a[$i] = $a{$q,$i};
}

# 乱数による答の並べ替え
for($i=0;$i<5;$i++){
    $ind[$i] = $i;
}
for($i=0;$i<5;$i++){
    $j = $i + int(rand(5-$i));
    $tmp = $ind[$i];
    $ind[$i] = $ind[$j];
    $ind[$j] = $tmp;
}

$s = "$q{$q}<br>\n";
for($i=0;$i<5;$i++){
    $_ = $a[$ind[$i]];
    s/././;
    $j = 63879+$i;
    if($ind[$i] == 0){ $ans = $i; }
    $s .= "&#j;$_<br>\n";
}
$s .= "<br>" x 5 . "\n";
$j = 63879+$ans;
$_ = $a[$ind[$ans]];
s/\t//;
$s .= "&#j;$_<br>\n";
$s .= "<a href=\"5tq.cgi\">次の問題</a><br>\n";

print <<EOF;
Content-type: text/html

<html>
<body>
$s
</body>
</html>
EOF
```