インターフェイスの街角– i モードを活用しよう!

増井 俊之

1 月号で、NTT DoCoMo の i モード端末の活用法を紹介しました。i モードブームはその後もとどまるところを知らず、i モードを使いこなすための雑誌記事や書籍が次々と現れています。

最近は端末自体の機能も多彩になり、電卓や予定表など、PDA のような使い方のできる機種もあります。とはいえ、蓄積できる情報量や操作性などの点ではまだまだノート PC や PDA にはおよびません。現時点では、i モード本体のこういった機能を使うよりも、i モード端末を UNIX マシンのリモート入出力装置として活用するほうがいろいろとおもしろい応用が考えられるでしょう。

そこで、今回は i モードのちょっと変わった使い方を紹介することにしました。

個人情報管理への応用

i モード端末の PDA 的な機能に加え、i モード向けのグループウェアのようなサービスを提供する企業も増えてきました。しかし、これらのサービスはメーリングリストの作成や予定表/名簿の管理といった最大公約数的なアプリケーションが中心で、現実の仕事にどの程度役立つかは現時点では未知数です。

重要な情報の管理を外部のアプリケーションサービス・プロバイダ任せにするのはやはりためらわれますし、個人的な情報については手許の PC に格納しておくのが普通だと思います。このような現状をみると、重要な情報の蓄積・管理という点では i モード端末はそれほど役立たないような気がします。

私は、情報を参照できなかったがために、以下のような失敗をすることがよくあります。

- おもしろそうな本についてメールで教えてもらったのに、書店でタイトルや著者名などを思い出せなかった。
- 秋葉原に行ったものの、何を買うはずだったか忘れた。
- 会員番号を忘れたため割引購入ができなかった。

おもしろそうな本や購入するつもりだった PC パーツの情報、会員番号などが PC 上のファイルのどこかに書いてあることは確実です。しかし、当然のことながら、PC が手許になければこれらの情報は利用できません。

最近の一般的な考え方からいえば、こういった個人的情報は PDA で扱うべきなのかもしれません。しかし、PDA を使って何千通ものメールのなかから求める情報を検索するのは非現実的ですし、とくに重要とは思えないものも含むあらゆるデータを、デスクトップ計算機と PDA とのあいだでつねに同期させるのは無駄だと思う人が多いのではないでしょうか。結果として、求める情報を PDA のなかから探しだす確率はかなり低くなってしまいます。

5 月号で、個人用情報管理システム Q-Pocket を紹介しました。これは、情報をカテゴリー分けすることなく超整理法的に管理し、高速曖昧検索によって内容を検索するシステムです。Q-Pocket はワークステーションや PC 上での個人情報管理にはたいへん便利ですが、情報の量はあっという間に増えるので、PDA のような記憶容量に制約のある機器と完全にデータを同期させるのは難しいという問題がありました。

ただし、Q-Pocket は検索サーバーとクライアントに分けて実装されているため、クライアント部だけを i モード端末から操作できるようにすれば、十分実用的なデータ検索が可能になります。i モード端末から Q-Pocket を用いた検索ができるのなら、上記のような問題も解決するで

リスト 1　検索・整形プログラム qpocket

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

use IO::Socket;

# Web公開ディレクトリ
$outdir = "/public/www";
# 検索結果リストファイル
$outfile = "$outdir/list.html";
# Q-Pocketデータ
$PIMDIR = "/user/masui/PIM";

sub id2filename {
    local($id) = @_;
    $id =~ /(....)(..)(..)(......)/;
    "$PIMDIR/$1/$2/$3/$id.pim";
}

# Query文字列の取得
while(<>){
    last if /^$/;
}
while(<>){
    chop;
    $query .= "$_ ";
}

# Q-Pocketサーバーと接続して検索実行
$remote = IO::Socket::INET->new(
    Proto    => "tcp",
    PeerAddr => "localhost",
    PeerPort => "5557",
    )
    ||die "Cannot connect\n";

print $remote "0 $query \n";
$_ = <$remote>;
s/^1//;
s/\r/\n/g;
chop;

# 検索結果をHTMLにフォーマット

open(out,"> $outfile");

print out <<EOF;
<html>
<head><title>検索結果</title></head>
<body bgcolor=white>
<font size=1>
<a href="mailto:qpocket\@my.mail.server">メールで検索</a>
<ul>
EOF

@result = split(/\n/,$_);
$resno = 1;
for (@result){
    s/\[(.{14})\]\s+//;
    $id = $1;
    $title = $_;
    $item = "res$resno.html";
    $src = &id2filename($id);
    $dst = "$outdir/$item";
    open(src,$src);
    open(dst,"> $dst");
    print dst "<html><font size=1><pre>\n";
    while(<src>){
            print dst $_;
    }
    close(src);
    print dst "</pre></html>\n";
    close(dst);
    print out "<li> <a href=\"$item\">[$id]<br>$title\n";
    $resno++;
    last if $resno > 10;
}

print out <<EOF;
</ul>
</body>
</html>
EOF
```

しょう。

メールによる i モードからの情報検索

今回は、メールで個人情報の検索をおこない、その結果をブラウザで閲覧する方法を実験してみました。

まず、/etc/aliases に以下のようなエントリを用意します。

```
qpocket: "| /user/masui/bin/qpocket"
```

次に、リスト 1 のプログラムを用意し、メールメッセージをキーワードとして Q-Pocket サーバーから検索します。そして、検索結果を HTML ファイルとして i モードから参照できるディレクトリ（ここでは/public/www を仮定）に置きます。

たとえば、〝restaurant〟というメッセージを含むメールを qpocket@my.mail.server 宛に送ると、リスト 1 のプログラム qpocket が起動され、指定されたキーワードで Q-Pocket サーバーを検索します。その結果、個人情報データをもとに以下のような検索結果表示ページが作成され、i モードから参照可能になります。

メールで検索

- [20000910200537]
  山彦

- [20000831151822]
  静雨庵
- [20000831151804]
  蔚
- [20000308170049]
  New restaurant
- [20000110010613]
  たべものや Kuishin坊
- [20000110005131]
  魚福

このように、メールサーバーと Q-Pocket サーバー、WWW サーバーを組み合わせて検索要求の受付け、検索の実行、表示の処理をおこなえば、非力な i モード端末でもデスクトップ計算機と同等の検索が可能になります[1]。

着メロで遊ぶ

いわゆる〝着メロ〟関連のサイトは、i モードで大きな人気を集めているようです。着メロがこれほどまで流行るとは思いませんでしたが、周囲でいろいろな曲が和音で鳴っているのを耳にすると、気の利いた着メロの 1 つくらい登録しておきたくなります。

しかし、多くの着メロサイトで提供されているのは、たいていは最近のヒット曲やひと昔前のアニメの主題歌などばかりで、かならずしも自分の好きな曲がみつかるとはかぎりません。いずれ著作権問題なども出てくるでしょうから、今後はフリーで提供される着メロは少なくなってきそうです。自力で手軽に着メロを作ったり送ったりできれば、世間の人気に左右されずに自分の好きな着メロを楽しめます。

i モード着メロのデータ形式

最近の i モード端末は、Web ページから直接着メロデータをダウンロードして演奏、登録できるようになっています。着メロ用のデータは、〝.mld〟という拡張子の〝MLD 形式〟のバイナリデータです。

Web ページ上では、HTML の A タグを使い、

```
<a href="mysong.mld">MySong</a>
```

のように MLD 形式のデータを指定すれば、i モード端末から直接 mysong.mld をダウンロードできます。

1 このようにして検索したファイルは、誰でもどこからでも閲覧できます。したがって、自分以外は知らない URL を使ったり、検索結果をすぐに消去するといった工夫が必要になります。

また、以下のようにして MLD 形式のバイナリデータを Base64 でエンコーディングし、先頭に〝--B:M〟という行を付ければ、i モード端末にメロディデータをメールとして送れます。

```
% echo '--B:M' ; mimencode mysong.mld | \
mail someone@docomo.ne.jp
```

つまり、MLD 形式のデータさえ作れれば、Web ページやメールでメロディデータを自由にやりとりできることになります。

ストトン記法による着メロデータ作成

MLD データ形式は一般には公開されていないようですが、ユーザーによる解析が進み[2]、Windows などで使える変換ツールが数多く公開されているため、おおまかな構造は推測できます。公開されている変換ツールをみると、MML (Music Macro Language) と呼ばれるテキストベースの音楽記述言語を使用するものや、GUI で鍵盤や楽譜を操作してメロディを入力するものが多いようです。今回は、1999 年 4 月号で紹介した、直感的な〝ストトン記法〟を着メロデータに変換する方法を紹介します。

ストトンシステムは、Perl を用いて楽曲を直感的に記述するシステムです。専用言語による特殊な表現方法ではなく、汎用的なプログラミング言語である Perl を利用します。たとえば、〝&play(”ドレミファ”)〟という関数を実行すれば〝ドレミファ〟という音が、〝&play(”スットントン”)〟を実行すれば、それに似た音のドラムが鳴るといったシステムです。

ストトンシステムは、専用の音楽記述言語を使うのではなく、Perl のライブラリとして MIDI を制御するため、Perl の変数や繰返し制御文などをそのまま演奏に利用できます。プログラムのエラーメッセージを MIDI で表現するといったことも簡単です。

プログラムの例

ストトンシステムでは、プログラムをそのまま音読できるようにするために〝ド〟や〝レ〟のような音名をそのまま使います。たとえば、ピアノとベースで「どんぐりころころ」を演奏させるプログラムはリスト 2 のようになります。

2 http://tokyo.cool.ne.jp/ittake/mld/mld.html など

リスト 2　ピアノとベースで「どんぐりころころ」

```
#!/usr/local/bin/perl
require 'midi.pl';
&prologue;
&inst(1,'Piano');
&inst(2,'Bass');
&channel(2);
&play("__");  # ベースを2オクターブ下げる
&parbegin;
    &channel(1); # ピアノチャンネル
    &serbegin;
        &play("ソッミミファミレド ");
        &play("ソッミミレーーッ");
    &serend;
    &channel(2); # ベースチャンネル
    &serbegin;
        &play("ミッドドレレソソミミドドソーーッ");
    &serend;
&parend;
&parbegin;
    &channel(1);
    &play("ミミソソララッラ^ドド_ミミソーッッ");
    &channel(2);
    &play("ドド ミミファファド ファレレララソーッッ");
&parend;
&epilogue;
```

リスト 3　ストトン表記版 MLD データ

```
#!/usr/local/bin/perl
require 'mld.pl';
&prologue;
&version("0100");     # バージョン番号
&date("20000920");    # 日付
&copyright("Masui"); # 作者
&title("Donguri");    # タイトル
&inst(1,4);           # チャンネル1の音色指定
&inst(2,5);           # チャンネル2の音色指定
&unitlen(15);
&channel(2);
&play("_");  # ベースを1オクターブ下げる
&parbegin;
    &channel(1); # ピアノチャンネル
    &play("ソッミミファミレドソッミミレーーッ");
    &channel(2); # ベースチャンネル
    &play("ミッドドレレソソミミドドソーーッ");
&parend;
&parbegin;
    &channel(1);
    &play("ミミソソララッラ^ドド_ミミソーッッ");
    &channel(2);
    &play("ドド ミミファファド ファレレララソーッッ");
&parend;
&epilogue;
```

ストトン表記で MLD 形式データを作成するには、midi.pl の代わりに mld.pl を使います（リスト 3）。リスト 2 とほとんど同じですが、MLD 形式では曲のタイトルや日付などを含めることもできます。

このプログラムを実行すると、以下のように MLD 形式のテキスト表現が出力されるので、これを txt2mld という Perl プログラム（リスト 4）でバイナリ形式に変換します。

```
% donguri | more
{melo} <<<224>>>
<<51>> <0x01> <0x01> <0x01>
{vers} <<4>> {0100}
{date} <<8>> {20000920}
{copy} <<5>> {Masui}
{titl} <<7>> {Donguri}
{trac} <<<163>>>
<0x00> <0xff> <<0xe044>>
<0x00> <0xff> <<0xe142>>
<0x00> <0xff> <<0xe085>>
<0x00> <0xff> <<0xe182>>
<0x00> <0x00> <0x00>
<0x00> <0x62> <0x0f>
<0x00> <0x93> <0x0f>
        ......
% donguri | txt2mld > donguri.mld
% (echo ''; echo '--B:M' ; donguri | txt2mld
 | mimencode) | mail someone@docomo.ne.jp
```

ストトン/MLD 変換プログラム

ストトン/MLD 変換は、末尾のリスト 5 のように比較的簡単に実現できます。

ストトンシステムでは、いわゆる〝打ち込み〟による入力とは異なり、メロディが計算機によって自動的に生成されます。たとえば、〝きょうのメロディ〟を自動作曲させたり、受け取ったメールを曲に変換したりと、いろいろ変わった使い方も考えられるでしょう。

システム管理への応用

i モード端末は、システム管理用の端末としても注目を集めています [1]。これまでにも、異常が発生した場合などに管理者のポケベルに連絡が送られるようなシステムはありましたが、あまり複雑な作業はおこなえませんでした。しかし、i モード端末であれば、管理者に異常発生を知らせるメールを送ったり、端末側からシステムの状態をモニターしたりすることも簡単です。

着メロ生成システムと組み合わせると、〝ディスク容量が少なくなってきたら悲しそうな音楽が送られてくる〟など、システム管理を楽しくする工夫もできそうです。

リスト 4　txt2mld

```
#!/usr/local/bin/perl
# txt2mld
while(<>){
  chop;
  $r = '';
  while($_ ne ''){
    if(s/^<(0x)?([\da-fA-F]+)>//){
      ($h,$c) = ($1,$2);
      $c = hex($c) if $h;
      $r .= pack("C",$c);
    }
    elsif(s/^<<(0x)?([\da-fA-F]+)>>//){
      ($h,$c) = ($1,$2);
      $c = hex($c) if $h;
      $r .= pack("n",$c);
    }
    elsif(s/^<<<(0x)?([\da-fA-F]+)>>>//){
      ($h,$c) = ($1,$2);
      $c = hex($c) if $h;
      $r .= pack("N",$c);
    }
    elsif(s/^\{([^\}]+)\}//){
      $r .= $1;
    }
    elsif(ord($_) >= 0x80){
      s/^..//;
    }
    else {
      s/^.//;
    }
  }
  print $r;
}
```

1/2AD スペース
(ノンブル段階で小口寄りに)

おわりに

i モード端末は、音声も画像も扱える小型で安価な通信システムとしてまだまだおもしろい応用が考えられそうです。現在のところ、音や画像を活用したゲームはないようですが、今後は増えてくるでしょう。

今回紹介したように、i モード端末は PDA に必要な機能の多くを備えていますし、通信機能については一般的な PDA よりはるかに優れています。PDA の通信機能が進化するのか、それとも i モード端末の入力機能が進化するのかなど、今後の動向から目が離せません。

（ますい・としゆき　ソニー CSL）

[参考文献]

[1] 若居和直、小池英樹「携帯電話による遠隔ネットワーク監視システム」、マルチメディア、分散、協調とモバイル (DICOMO 2000) シンポジウム 論文集、pp.415–420、2000 年

リスト 5 mld.pl

```
%note = ( # 音名と音高番号の対応
    'ド', 27, 'レ', 29, 'ミ', 31, 'ファ', 32,
    'ソ', 34, 'ラ', 36, 'シ', 38, 'ッ', -1,
);

$notepat = join('|',keys(%note));
$level = 0;         # ネストレベル
$env[$level] = 's'; # par / ser
$channel = 1;
$unitlen = 100;

sub unitlen{    # 演奏単位時間の設定
        $unitlen = $_[0];
}
sub inst {
        local($ch,$inst) = @_;
        &add(sprintf("0 0 C e0%02x",
                ($ch << 6) + ($inst & 0x3f)),0);
        &add(sprintf("0 0 C e1%02x",
                ($ch << 6) + 0x02 + (($inst & 0x40)>>6)),0);
}
sub version { $versiondata = $_[0]; }
sub date { $datedata = $_[0]; }
sub copyright { $copydata = $_[0]; }
sub title { $titledata = $_[0]; }
sub channel { $channel = $_[0]; }
sub parbegin { &begin('p'); }
sub serbegin { &begin('s'); }
sub begin {
        $level++;
        $env[$level] = $_[0];
        $out[$level] = '';
        $len[$level] = 0;
}
sub serend { &parend; }
sub parend {
        $level--;
        &add($out[$level+1],$len[$level+1]);
}
sub add { # 既存データに新しいデータを追加
        local($buf,$len) = @_;
        if($env[$level] eq 's'){ # データを連接
                $out[$level] .= &timeshift($buf,$len[$level]);
                $len[$level] += $len;
        }
        else { # データをマージ
                $out[$level] = &timesort($out[$level].$buf);
                $len[$level] = ($len[$level] > $len ?
                                $len[$level] : $len );
        }
}
sub play {
        local($_) = @_;
        local($t,$buf,$note,$notename,$ts);
        $ts = $t = 0;
        while($_ ne ''){
                if(s/^_//){ $offset[$channel] -= 12; next; }
```

```
                if(s/^\^//){ $offset[$channel] += 12; next; }
                if(s/^ー//){ $t += $unitlen; next; }
                if(s/^($notepat)([#b]?)//){
                        $notename = $1;
                        $sharpflat = $2;
                        if($t > 0 && $note > 0){
                                $buf .= sprintf("$ts $t $channel $prevnote\n");
                        }
                        $note = $note{$notename};
                        $note += ($sharpflat eq '#' ? 1 :
                                  $sharpflat eq 'b' ? -1 : 0);
                $ts = $t;
                $prevnote = $note+$offset[$channel];
                }
                elsif(s/^([\x80-\xff])(.)//){ } # ドレミ以外の漢字
                else { s/^.//; }
                $t += $unitlen;
        }
        if($t > 0 && $note > 0){
                $buf .= sprintf("$ts $t $channel $prevnote\n");
        }
        &add($buf,$t);
}
sub bytime {
        local($ta,$tb);
        $a =~ /^(\d+)\s/;
        $ta = $1;
        $b =~ /^(\d+)\s/;
        $tb = $1;
        $ta <=> $tb;
}
sub timesort { # イベントを時刻順にソート
        local($s) = @_;
        local(@a);
        @a = split(/\n/,$s);
        @a = sort bytime @a;
        join("\n",@a) . "\n";
}
sub timeshift { # イベント発生時刻をシフト
        local($s,$t) = @_;
        local(@a,$i,$s1,$s2);
        @a = split(/\n/,$s);
        for($i=0;$i<=$#a;$i++){
                $a[$i] =~ /^(\d+)\s+(\d+)\s+(.*)$/;
                $s1 = $1; $s2 = $2; $s3 = $3;
                $a[$i] = ($s1+$t) . ' ' .  ($s2+$t) . " $s3";
        }
        join("\n",@a) . "\n";
}
sub prologue { }
sub epilogue {
        @out = split(/\n/,$out[0]);
        $s = '';
        $prevt = 0;
        for (@out){
                ($ts,$te,$ch,$note) = split;
                $dt = $ts - $prevt;
                $prevt = $ts;
```

```
                if($ch eq 'C'){
                        $s .= sprintf("<0x%02x> <0xff> <<0x%s>>\n",$dt,$note);
                }
                else {
                        $s .= sprintf("<0x%02x> <0x%02x> <0x%02x>\n",$dt,
                                (($ch << 6) + $note),$te-$ts);
                }
        }
        $trackdata = $s;
        $trackdatalen = &len($trackdata);
        $trackheader = "{trac} <<<$trackdatalen>>>\n";
        $trackblock = $trackheader . $trackdata;

        if($versiondata){
                $versiondatalen = length($versiondata);
                $versionblock = "{vers} <<$versiondatalen>> {$versiondata}\n";
        }
        if($datedata){
                $datedatalen = length($datedata);
                $dateblock = "{date} <<$datedatalen>> {$datedata}\n";
        }
        if($copydata){
                $copydatalen = length($copydata);
                $copyblock = "{copy} <<$copydatalen>> {$copydata}\n";
        }
        if($titledata){
                $titledatalen = length($titledata);
                $titleblock = "{titl} <<$titledatalen>> {$titledata}\n";
        }

        $padding = "<0x01> <0x01> <0x01>\n";
        $prefix = $padding . $versionblock . $dateblock .
                $copyblock . $titleblock;
        $prefixlen = &len($prefix);
        $headerblock = "<<$prefixlen>> $prefix";

        $melodydata = $headerblock . $trackblock;
        $melodydatalen = &len($melodydata);
        $melody = "{melo} <<<$melodydatalen>>>\n" . $melodydata;

        print $melody;
}

sub len {
        local($_) = @_;
        local($len) = 0;
        while($_ ne ''){
                if(s/^<<<[^>]+>>>//){ $len+=4; }
                elsif(s/^<<[^>]+>>//){ $len+=2; }
                elsif(s/^<[^>]+>//){ $len++; }
                elsif(s/^\{(.[^\}]*)\}//){ $len += length($1); }
                else { s/^[\x00-\xff]//; }
        }
        $len;
}
1;
```