
インターフェイスの街角 - Baby Face のインターフェイス

増井 俊之

最近、携帯電話や携帯型計算機など、画面の小さい機器がよく使われるようになってきました。2000 年 1 月号に書いたように、これらの機器は "Baby Face" と呼ばれています。Baby Face では、大きな画面を備えた計算機には不要なさまざまな工夫が必要になります。今回は、Baby Face を利用するための技術をいくつか紹介し、これを携帯電話へのメール転送に応用する方法を解説します。

小さな画面を活かすには

小さな画面で多くの情報を扱えるようにするには、以下のような方法が考えられます。

- 表示内容を縮小する
小さなフォントを使ったり図形を縮小すれば、小さな画面にも多くの情報が表示できます。たとえば、2000 年 10 月号で紹介した ClearType のような技術を使うと、小さなフォントでもきれいに表示されます。
また、Web ページやメールの内容はすべてが重要とはかぎらないので、必要な箇所だけを表示したり、表示内容を圧縮する方法も有効です。
- 情報を重ねて表示する
異なる種類の情報は、同じ場所に重ねて表示できる場合があります。たとえば、テキストの上に半透明なアイコンを重ねて表示すれば、場所を節約できます。
- ユーザーの操作により表示を変える
一度に表示できる情報の量はどうしても限られますが、ユーザーの操作により異なる情報が見えるようにすれば、それだけ多くの情報を提示できるようになります。画面上で長いリストを見るときはスクロールバーがよく

使われますし、最近、ズーム・インターフェイスによって画面を拡大/縮小する方法もさかんに研究されています。マウスをクリックしたときだけ表示されるメニューも、小さな画面を有効に活用するための技術の 1 つといえます。あるいは、見る部分や方向によって見える内容が変わるようにするのも有効かもしれません。

- 時分割で異なる情報が表示されるようにする
電光掲示板や動画/アニメーションを使えば、小さな場所にも多くの情報を表示することができます。

いろいろな手法

上記の技法を組み合わせたいろいろなシステムが開発されていますが、それらのなかから、やや毛色の変わったものをいくつか紹介します。

GUI 部品の重ね合せ

日本電気の神場知成氏は、半透明なアイコンをテキストなどと重ねて表示する場合のインターフェイス手法に関する研究をおこなっています [1]。

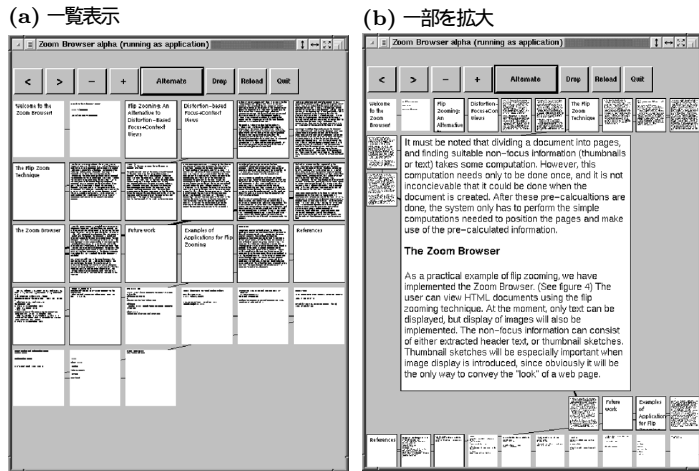
操作のアイコンが表示されていても、その下のテキストを読めるので、小さな画面領域を有効に活用できます。

通常のテキストならこれで問題はありますが、Web ページのようにリンクを含むテキストの場合は、アイコンとリンクが同じ場所に重なってしまうため、どちらかが操作不能になってしまいます(図 1 では、画面の下部でテキスト中のリンクとアイコンが重なっています)そこで、神場氏のシステムでは、GUI 部品が重なっているときは、操作時間によって優先度が切り替えられるようになっています。たとえば、アイコンとリンクが重なっている場所を

図 1 透明アイコンを重ねて表示した画面



図 2 Flip Zooming



一瞬タップするとアイコンが選択されますが、長いあいだ押し続けているとアイコンを“突き抜けて”下側にあるリンクが選択されます。

Flip Zooming

Flip Zooming は、スウェーデンの Viktoria Institute の Lars Erik Holmquist 氏¹が開発した小型端末用の表示手法です。

1つの画面に多くの画像やページを並べて表示しようとすると、1つ1つの画像がかなり小さく見づらくなります。一覧情報とともに、一部を詳しく表示できるようにするために、“非線形ズームング”という手法が使われることがあります²。非線形ズームング・システムは、一部だけを連続的に拡大して表示できるという特色をもっていますが、操作に対して画面がぐにやぐに動くような印象があり、かならずしも使いやすいとはいえません。

Flip Zooming は、複数の画面のうちの1つだけを大きく表示し、残りは小さなサイズでその周辺に配置する手法です(図2)

図2-aは、Flip Zooming用のテキストブラウザによる一覧表示画面です。図2-bでは、文書の13ページ目の一部が拡大されています。この場合、周辺の縮小されてい

るページの内容はほとんど分かりませんが、全体の雰囲気は掴めるのではないのでしょうか。

メールの圧縮表示

メールを携帯電話やポケベルなどに転送するとき、主要な部分に絞って送れば小さな画面でも必要十分な情報が得られます。日本電気の坂上秀和氏は、携帯電話などへメールを転送する際、メッセージを劇的に圧縮する手法を提案しています[2]。

通常のメールメッセージには、一般に冗長な情報が数多く含まれています。たとえば、先頭部分に、

「ソニーの増井です」

などと書くことがよくあります。しかし、これは From: 行を見れば明らかなので、わざわざ書く必要はないかもしれません。さらに続けて、

「ご無沙汰しております」

などと書く人も多いでしょう。これも、ご無沙汰していることが明白であれば不要な情報かもしれません。メールの末尾によくある、

「よろしくお願いします」

といった文言もとくに重要ではないので、読み飛ばしても問題はないはずです。要件さえ伝わればよい場合、このような決まりきった表現は思いきってすべて省略してもよいのではないのでしょうか。完全に省くのが問題なら、「あり

1 <http://www.viktoria.informatik.gu.se/~leh/>

2 非線形ズームング・システムについては、<http://www.cs.indiana.edu/hyplan/tkeahey/tkeahey/research/nlm/nlm.html> に詳しい解説があります。残念ながら、ひろく使われているシステムはまだないようですが、Mac OS X のデスクトップ下部に表示される“Dock”のようなものといえればかなりやりやすいかもしれません。

がとうございました」を「TNX」と略したり、「申し訳ありません」を「SRY」と略したりすることもできます。

お互いによく知っている固有名詞などは、略記しても意味が通じることがあります。たとえば、私にとっては「ソニーコンピュータサイエンス研究所」と書くまでもなく、「SonyCSL」や「CSL」でも十分です。場合によっては、引用部をすべて削ったり、シグネチャを省略することも可能でしょう。

このように、とくに重要ではない文字列を省略したり、文字列を省略表記に置き換えたりすれば、不要な部分をほとんど切り捨てて、メッセージをかなり短縮できます。

坂上氏らの論文 [2] には、図 3 のような例が紹介されています(メールアドレスは一部変更してあります)。上記のような文字列置換のほか、全角文字を半角文字に変換したり、From:や To:などの表現にも工夫を加えて、ポケベルへの転送が可能な半角 100 文字以内でもなんとか意味がとれるようにしています。

携帯電話へのメール転送システム

坂上氏らの手法を応用して、メールを圧縮して携帯電話に転送するシステムを作ってみました。重要と思われる個人向けのメールだけを圧縮し、携帯電話に転送します。

転送の設定

sendmail や qmail などでは、ホーム・ディレクトリの .forward ファイルにメールの転送先アドレスやフィルタ・プログラムを記述しておけば、受信したメールについていろいろな処理をおこなうことができます。今回は、受け取ったメールはすべて保存し、重要と思われるメールだけを圧縮して携帯電話に送ることになります。

メールのフィルタリングと転送には procmail³を使います。procmail は、sendmail などの転送機能を用いてメールを振り分けることが可能なシステムです。procmail では、Subject:や From:などの行を対象にパターン・マッチングをおこない、マッチしたメッセージを別のアドレスに転送したり、別のフィルタ・プログラムに渡したりすることができます。procmail には数多くの機能がありますが、設定ファイルの記述方式が分かりにくい

3 <http://www.procmail.org/>

図 3 メッセージ圧縮の例

```
(a) 変換前のメッセージ
Subject: OB合宿について
To: sakagami@ccm.cl.nec.co.jp
Cc: okano@xxx.or.jp
From: okano@yyy.zzzz.co.jp

岡野@C&Cメディア研究所です。
> サッカー部OB会の合宿の御案内
>
> 日時: 1998年 8月1日~8月2日
> 場所: 検見川総合合宿場
>
> OBの方は、8月2日に試合を行いますので、
> 是非、当日の午前9時までにお越しください。
申し訳ありません。
明日は午前中に急用が入ったため、午後には駆けつけます。
よろしくお願いします。

(b) 変換後のメッセージ
野人>俺(会長)[OB合宿について]岡野
@CCMです。明日はAM中に急用が入った
ため,PMには駆けつけます。
```

め、さまざまな代案が考えられています⁴。たしかに、複雑な振り分け処理をしようとする面倒ですが、単純なパターン・マッチングや転送程度であれば、さほど苦労せずに使えます。

まず、~/forward に図 4 のような記述を加えます。届いたメールをすべて procmail の標準入力に送るように指定しているだけですが、sendmail との関連で、シェルのセパレータ文字 (IFS : Internal Field Separator) を空白文字に設定したり、エラー番号を指定したり、各自の ~/forward ファイルがユニークであることを保証したりする必要があるので。

procmail の動作は ~/procmailrc で指定します。図 5 のように記述しておけば、すべてのメッセージを標準的な方法で保存したあと、重要なメッセージだけをフィルタに通して携帯電話に送ることができます。

メッセージの圧縮

procmail を用いた Subject:や From:のマッチングにより、重要なメールだけをフィルタリングし、その後述の方式でテキストを圧縮します。

メッセージの圧縮は、compressmail プログラムでお

4 <http://www.itknowledge.com/tpj/feature/tpj0502-0002.html> や <http://www.namazu.org/~satoru/mail-classify/> などで、procmail を使わずにメールを振り分ける手法が紹介されています。

図 4 ~/.forward の記述例

```
"|IFS=' ' && /usr/local/bin/procmail || exit 75 # masui"
```

図 5 ~/.procmailrc の記述例

```
# SPAMを撃退
:0:
* ^From:.*spam@spam.spam
/dev/null
# すべてのメールを保存し、その後の処理は
# コピーに対して適用する
:0c:
$DEFAULT
# メールングリストからの配信と思われるものを除く
:0:
* ^Subject:.*\[
/dev/null
# メールマガジンなどを除く
:0:
* ^From:.*mag2
/dev/null

# エラーメールなどを除く
:0:
* ^Subject:.*Returned.*mail
/dev/null
# それ以外のメールを圧縮する
:0 fw:
| /user/masui/bin/compressmail
# 圧縮されたテキストを携帯電話に転送する
:0 A:
! iModeAddr@docomo.ne.jp
```

図 6 compressmail プログラム(一部)

```
#!/usr/bin/perl

sub jis2ujis { ..... }
sub z2h { ..... } # 全角半角変換

sub deldup { # 連続する記号を省略
    local($_) = @_;
    1 while s/([\d\w\x80-\xff])\1\1\1\1/;
    1 while s/([\x80-\xff].)\1\1\1\1/;
    $_;
}

@pats = (
    '(いつも)?お世話にな(ってお)?ります', 'OSW',
    '(宜|よ)しくお願|申しあげます', 'YRS',
    'よろしく', 'YRS',
    'ありがとうございます', 'TNX',
    'ありがとうございます', 'TNX',
    'ありがとう', 'TNX',
    'す(み)い(い)ません', 'SRY',
    '申し訳ない', 'SRY',
    '申し訳ございません', 'SRY',
    '.....',
    'できません', '不能',
    'だめ', 'NG',
    'いけない', 'x',
    '非常に', '超',
    'とても', '超',
    '嬉しい', ' ',
    '.....',
);

sub out {
    local($_,$i) = @_;
    if($_ ne ''){
        for($i=0;$i<=$#pats;$i+=2){
            $from = $pats[$i];
            $to = $pats[$i+1];
            s/$from/$to/g;
        }
    }
}

print &jis2ujis($_),"\\n";
$line = '';
}
exit if $lineno++ > 50;

# ヘッダ処理
while(<>){
    chop;
    $_ = &jis2ujis($_);
    &out($_) if /^From:/;
    &out($_) if /^Subject:/;
    last if /^\\s*$/;
}
print "\\n";

# 本文処理
while(<>){
    chop;
    s/^\\s*//;
    $_ = &jis2ujis($_);
    $_ = &z2h($_); # 全角 半角
    $_ = &deldup($_); # 連続する記号を省略
    if(/>/){
        &out($line);
        # 連続する引用行は1行目だけ表示
        &out($_) if $quotes++ == 0;
    }
    else {
        $quotes = 0;
        $line .= $_;
        # 空行または記号のみの行で改行
        if(/^\\s*$/ || /^[^\\s\\d\\x80-\\xff]*$/){
            &out($line);
        }
    }
}
&out
```

図 7 compressmail によるメッセージ圧縮

```

(a) 変換前のメッセージ
Subject: 御礼
From: masui@xxx.co.jp

先日は本当にありがとうございました。
今後ともよろしくお願ひ申し上げます。

-----
増井 俊之
ソニーコンピュータサイエンス研究所
〒141-0022 品川区東五反田3-14-13
Phone: 03-xxxx-xxxx
Fax: 03-xxxx-xxxx
E-mail: masui@xxx.co.jp
URL: http://www.csl.sony.co.jp/~masui/

(b) 変換後のメッセージ
Subject: 御礼
From: masui@xxx.co.jp

先日はマジTNX. 今後ともYRS.
--
増井

```

はびっくりしましたが、慣れればとくに問題はなさそうです(むしろ、普通のメッセージに「SRY が」などを書いてしまいそうなのが心配なほどです)

現在のところ、i モードのメールで受信可能な文字列は 1,000 文字に限定されています。PC などで作成したメールメッセージは 1,000 文字を超えるのが普通なので、メールを単純に転送すると、たいていはメッセージの一部しか受け取れません。しかし、このような圧縮手法を適用すれば、かなりの量の情報が受け取れるようになります。

“ありがとうございました”を THX と略しても違和感がなくなるのなら、“ありがとうございました==THX”という対応関係が頭のなかにできあがるので、テキスト入力システムで“THX”を“ありがとうございました”と変換しても抵抗はなくなりそうです。このような対応関係ができれば、入力も表示もコンサイスにまとめることが dkrynrrds(できるようになるでしょう)

全角 250 文字まででは?

こないます。文献 [2] の例では From:行なども圧縮の対象にしていたのですが、それでは返信などができなくなるので最低限のヘッダ情報は残すことにします。

本文に対し、compressmail では、

- 全角文字を半角文字に変換
- 同じ記号が連続するときは省略

という処理をおこなったあと、文字列を置換します。

compressmail プログラムを図 6 に、compressmail による変換前と変換後のテキストを図 7 に示します。

使用感など

compressmail ではごく単純な文字列置換をおこなっているため、部分文字列が不適切に置換されることがあります。たとえば、“すみません SRY”という置換規則では、“すみません”は“SRY が”に変換されてしまいます。こういった問題を根本的に解決するには、“すみせんが”を省略するなど、置換規則にもっと工夫を加えるべきかもしれません。しかし、こういった表現には比較的すぐに慣れるようで(個人差はあると思いますが)頭のなかに“すみせん SRY”という規則が入っていれば違和感なく読めます。NEC の人たちが、「SRY が」などというメッセージをやりとりしているのを初めて目にしたとき

おわりに

Baby Face 機器が普及し始めてから日が浅いこともあって、これらの機器用のインターフェイスの研究はまだ十分とはいえません。将来、計算機の性能がいくら向上しても、おそらく携帯型計算機の画面が小さいという点は変わらないでしょう。今後、ハードウェアやソフトウェアによって画面の小ささを克服するような優れたインターフェイス手法が開発されることを期待したいと思います。

(ますい・としゆき ソニー CSL)

[参考文献]

- [1] Tomonari Kamba, Shawn A. Elson, Terry Harpold, Tim Stamper and Piyawadee “Noi” Sukaviriya, “Using small screen space more efficiently”, In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI'96)*, pp.383-390, Addison-Wesley, 1996
- [2] 坂上秀和、神島和成「超小型画面向けメッセージ表示インタフェース」『インタラクティブシステムとソフトウェア VI』(安村通晃・編) 日本ソフトウェア科学会 WISS'98, pp.165-174、近代科学社、1998 年