

---

## インターフェイスの街角 (2)

### 動的曖昧検索

増井俊之

---

今回は「動的曖昧検索」の手法と文章入力への応用について解説します。

---

### 動的検索と曖昧検索

#### 動的検索

インターネットの普及により、検索をおこなう機会が増えてきました。grep や ls など、よく使われるコマンドの多くは検索の一種ですし、find やシェルのパターンマッチ、かな漢字変換の辞書検索などを含めると、日常的な作業のかなりの部分が検索に関係しているといえます。これらの検索作業では、検索条件を指定したあとで検索実行を指示することにより検索が開始される点が共通しています。ユーザーが明示的に指示しないかぎり、検索は始められません。

Web ページでの検索のように検索に時間がかかる場合などはこのほうが好都合ですが、十分高速でありかつ検索対象がはっきりしている場合は、検索条件の指定と同時に検索を実行するほうが便利ながよくあります。たとえば、Emacs の isearch (incremental search) では、検索実行を指示しなくても、検索文字列の指定と同時に検索が始まり、それにつれてカーソルが該当箇所に順次移動してくれます。また、Windows 用の電車経路探索プログラム「駅すばあと」<sup>1</sup>で駅名を入力する場合、「し」と入力すると同時に「品川」など「し」で始まる駅名がリストされるようになっているので、駅の読みをすべて入力しなくても目的の駅を選べます。

このような、検索条件の指定と同時に検索が実行さ

れ、結果が表示されるような方式は動的検索 (Dynamic Query)<sup>[2]</sup>と呼ばれています。

#### 直接操作と動的検索

近年の GUI (Graphical User Interface) は、いわゆる直接操作 (Direct Manipulation)<sup>[1]</sup>をサポートしています。画面上のアイコンやボタン、ウィンドウなどを実物と同じような感覚で操作するインターフェイスのことを直接操作インターフェイスと呼びます。

画面上の表示を実物と同じように見せるには、ユーザーが操作すると即座にその操作量に対応した反応が起こるようにすると有効です。金槌でも蛇口でも自動車でも、世の中のほとんどの道具は操作をするとすぐ反応が返るようになっていますから、計算機も同じようにすればよいわけです。たとえば、ウィンドウをドラッグするときは、マウスを動かすと同時にその移動量に対応した距離だけウィンドウが動くので、あたかも直接ウィンドウを持って動かしているような感覚が得られます。あらゆる GUI がこのような「連続的な」感覚をサポートしているわけではありませんが、計算機をより直感的に使えるようにするためには、連続性はきわめて重要な要素の 1 つです<sup>[5]</sup>。

連続的なシステムでは、比較的簡単に「可逆性」を実現できます。多くのウィンドウ・システムではクリックひとつでウィンドウを閉じたり開いたりできますが、たいいていは閉じるための操作と開くための操作は異なっています。このように非連続的な操作は非可逆的になりがちですが、ボタンなどで瞬時にウィンドウを閉じるのではなく、ズーム操作などによって連続的にウィンドウを縮小するインターフェイスでは、可逆的な操作でウィンドウを縮小したり、拡大したりすることができます。可逆的なインター

---

1 <http://www.val.co.jp/expworld/spec/expert.htm>

フェイスを採用すると、操作を間違えた場合でもつねに逆の操作をすればもとに戻せるので、直感的に扱いやすいシステムが作れるようになります。

動的検索は「ユーザーが検索条件を変えると、即座に対応して検索結果表示を変える」ものですから、直接操作インターフェイスによくマッチしています。ひと昔前の画面表示や検索の速度が遅かった計算機では動的検索の実現は困難でしたが、高速な計算機が手軽に使える現在では、資源をより富豪的<sup>2</sup>に用いて直接操作/動的検索を実行することにより、さらに使いやすいインターフェイスが構築できます。

## 曖昧検索

インターネットの検索エンジンや grep などの検索コマンドといった多くの検索システムでは、検索パターンに完全にマッチしないテキストは検索できません。たとえば、AltaVista で `masui tosiyuki` という文字列を検索しても、私のページはみつかりません ( toshiyuki となっているからです )、あるいは、英和辞書ファイルに対して、

```
% grep pitecan 辞書ファイル
```

としても、`ピテカントロプス (Pithecanthropus)` にはいきあたらないでしょう。

これでは不便なので、テキストが検索パターンと多少異なってもマッチングが成功したとみなす曖昧検索<sup>3</sup>をおこなうシステムが数多く提案されています。

agrep<sup>[3]</sup> は、そのような曖昧検索システムの 1 つで<sup>4</sup>、普通の grep と同様に使えるのももちろん、引数で曖昧度 ( 何文字までのミスマッチを許すか ) を指定することができます。

/usr/dict/words などの英単語ファイルに対して、引数を付けずに agrep を実行すると、普通の grep と同様に完全なパターンマッチをおこなうため、

2 高速な CPU や多量のメモリなどを惜みず使う `富豪的プログラミング` を提唱しています<sup>[4]</sup>。資源をけちってはいは、よいインターフェイスはなかなかできません。

3 シソーラスなどを利用して意味を考慮する方法も曖昧検索と呼ばれることがあります。ここでは単純な字面の比較による `approximate pattern matching` を指します。

4 ftp://ftp.win.or.jp/pub/misc/agrep-2.04.tar.gz などから入手できます。

```
% agrep masui words
%
```

のように何も出力されません。一方、引数 -1 を指定して 1 文字までの誤りを許容させると、

```
% agrep -1 masui words
massif
massive
mastic
mastiff
swimsuit
%
```

のように `masui` に近い単語のリストが得られます。

上の例では曖昧度をユーザーが明示的に指定しましたが、agrep に自動的に決めさせることもできます。-B (best match) オプションを指定すると、agrep は該当する行がみつかるまで曖昧度を増やししながら検索を繰り返し、パターンにもっとも近い行を出力します<sup>5</sup>。

```
% agrep -y -B masui words
massif
massive
mastic
mastiff
swimsuit
% agrep -i -y -B pitecan words
Aztecan
pecan
Pithecanthropus
pitman
%
```

このように、自動的に曖昧度を変化させながらパターンにもっとも近いものを探す手法は、外国語や固有名詞などの検索時にとくに便利です。

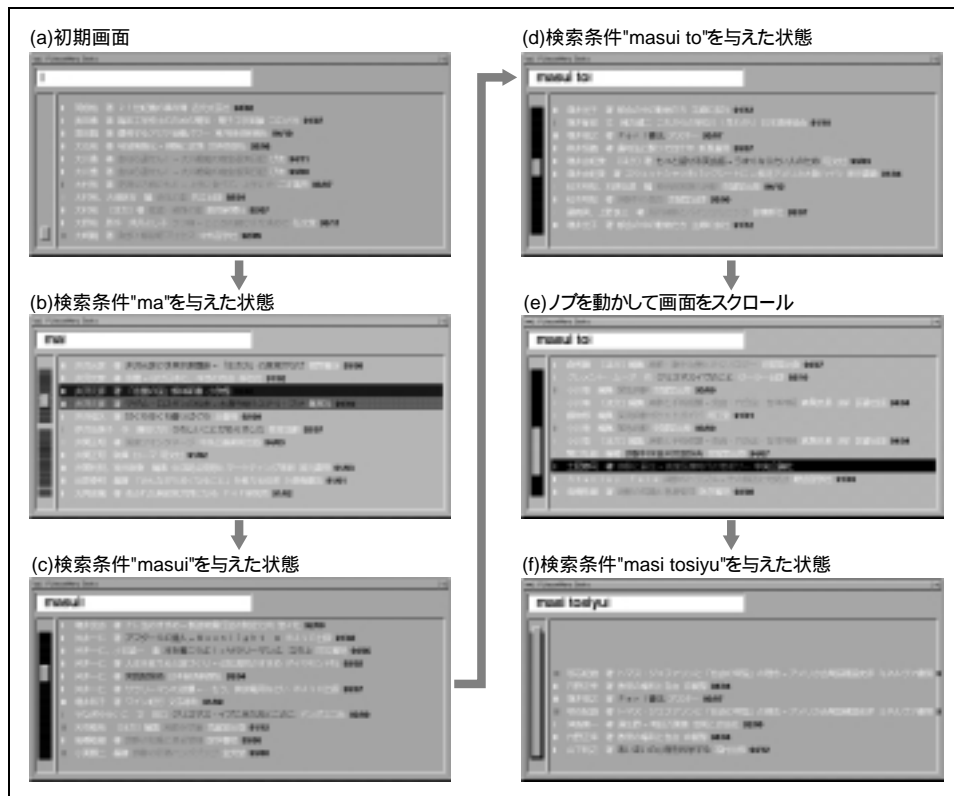
---

## 動的曖昧検索システム

検索条件の変化に応じてリアルタイムに検索する動的検索機能と、agrep のように曖昧度を自動的に調整する曖昧検索機能とを融合すれば、さらに便利な動的曖昧検索システムが構築できます。このようなシステムでは、ユーザーが条件を入力すると即座に検索結果が表示され、条件に正確に一致するデータがない場合には自動的に曖昧検索が実行されて、その結果が表示されます。

5 -y オプションを指定した場合、出力時にユーザーの確認をしません。このオプションを付けないと、agrep はマッチした行数をユーザーに知らせ、確認を求めているから出力します。

図 1 書籍検索システム



動的曖昧検索を用いたインターフェイスの一例として、私が SGI マシン上で開発した書籍検索システムを紹介します。

日本では年間数万点の書籍が発行され、たとえば電子ブック「本の探偵」(メディアパル)には過去5年間に出版された約30万点の書籍データが格納されています。30万ものデータを検索するのは大変そうですが、最近の高性能なPCやワークステーションには数十MBのメモリが載っていますし、CPUもきわめて高速です。したがって、すべてのデータをメモリ上に展開すれば、たとえ著者名や題名が不正確でも、動的曖昧検索を用いて即座に目的の書籍を検索できます。

書籍検索システムの最初の画面は、図1-aのようになっています。まだ検索条件を与えていないので、すべてのデータが一覧表示されています。左端のスクロールバーを使って、表示箇所を変えることができます。モノクロでは分かりづらいかもしれませんが、本の種類や価格などによ

り文字の色やアイコンに変化がつけられています。

スクロールバーを細かく動かしてすべての書籍データを眺めることもできますが、数十万行のリストを眺めるのは大変です。そこで、キーワード指定によるフィルタリングをおこないます。

検索条件として“ma”を与えると、画面はすぐに図1-bのように変化します。すべてのエントリのうち、“ma”という読みを含むものだけが表示されています。

続けて“sui”と入力すると、画面は図1-cのように変化し、著者名に“増井”を含む本や“麻醉”に関するものなどが表示されます。

スクロールバーの背景は、検索条件に該当する行の分布を表しています。全データのうち、読み“masui”にマッチしたエントリに対応する位置がスクロールバー上の灰色の横線で、マッチしていない部分が黒い横線で表現されているので、データ全体のうちどのくらいの部分が該当したかが一目でわかります。

さらに、空白文字に続けて“to”を指定すると、画面は図 1-d のように変化します。ここでは空白文字をワイルドカード指定としているため、リストに対して、

```
% grep 'masui.*to' booklist
```

を実行した場合と同様な結果が表示されています。

キーワードが指定されている状態でスクロールバーを動かすと、スクロールバーは条件に該当する行の上だけを移動します。したがって、マッチングの分布によって図 1-e のようにノブが伸び縮みます。

新たな検索条件として“masi tosiyu”を指定すると、マッチするデータがないため曖昧検索が実行され、これに近い本が検索結果として表示されます(図 1-f)

検索をおこなう場合、一般にはサーチエンジンによる検索や grep によるパターンマッチのようなキーワード検索と、Web ページのリンクから目的のものを探したり、長い文書をスクロールしながら目的とする箇所を探すようなブラウジングの 2 種類の手法が使われます。これらの手法は独立して用いられることが多く、キーワード検索と並行したブラウジングはできないのが普通です。たとえば図書館では、目録カードによるキーワード検索や書棚に並んだ本の背表紙を見ながらのブラウジングが可能ですが、両方を同時に実行することはできません。計算機上のプログラムでも、この両者を併用できる検索システムはあまりありませんが、上に紹介した書籍検索システムでは両者をうまく融合しています。おおまかなキーワードを指定すればすぐに結果が絞り込まれ、その結果に対してスクロールによりブラウジングができるので、どちらか一方だけを個別に実行するよりも、目的とする対象をはるかに容易に検索できます。

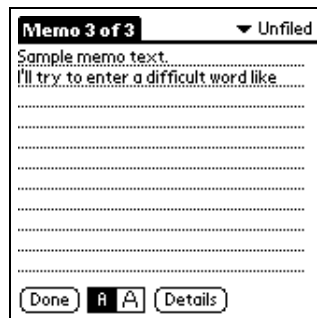
## 動的曖昧検索を用いた文章入力

動的曖昧検索はさまざまに応用できますが、文章の入力にも有効です。

### 検索操作と入力操作

検索と入力は一見まったく異なる操作のように思えますが、かな漢字変換で文章を入力する場合は、読みを漢字に変換するたびに変換辞書を検索した結果を使っています。

図 2 Pilot のメモ帳画面



あるいは、例文集を使って手紙を書くときなども、目的に近い例文を検索し、それを修正して使うわけですから、検索と入力の 2 つの作業が密接に関係しています。かな漢字変換の場合と同じように、動的曖昧検索の結果を入力文として利用すれば、入力作業の効率化が図れる可能性があります。

英語の文章を書いているときなど、綴りがあやふやで作業を一時中断せざるをえないことがよくあります。計算機を使っていなければ辞書を調べてから正しい綴りを書くでしょうし、計算機を使っているのなら、適当な綴りで文章を書いておき、あとでスペルチェッカーで修正する方法が一般的かもしれません。いずれの場合も単語の入力と検索は別々の作業ですが、動的曖昧検索を用いてこれらを融合することができます。

以降では、前回紹介した携帯情報端末 Pilot 上でこのような機能を実現した ASearch について説明します<sup>6</sup>。

### ASearch—動的曖昧検索 FEP

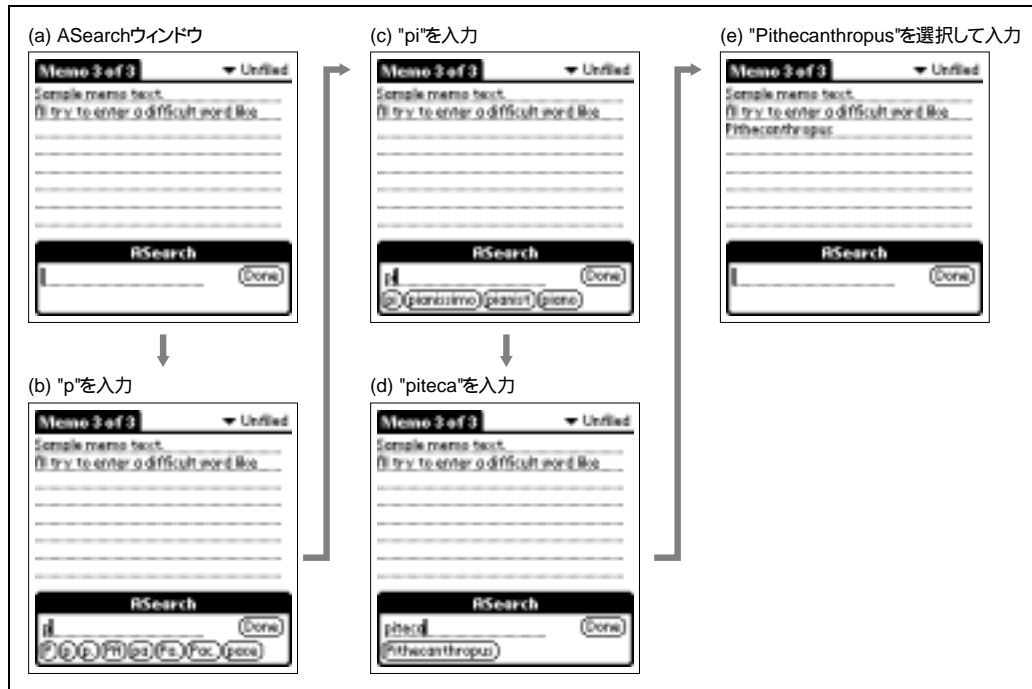
図 2 は、Pilot のメモ帳画面です。前号でも簡単に触れましたが、Pilot では画面下部の手書き入力領域に Graffiti という一筆書き方式で英数字を描いて文字を入力します。

ASearch は Pilot 上の Hack プログラム<sup>7</sup>で、ソフトウェア呼出し関数 SysKeyboardDialog() を置き換えます。HackMaster( Hack プログラムを管理するためのプログラム )に ASearch を登録し、ソフトウェア

<sup>6</sup> ASearch の詳細については次号で解説します。ASearch のソースプログラムなどは、<http://www.csl.sony.co.jp/person/masui/UnixMagazine/asearch.tar.gz> から入手できます。

<sup>7</sup> Hack とは、Pilot のシステム・ライブラリのトラップテーブルを書き換えて実現するプログラムです(詳細は前号を参照してください)

図 3 ASearch による入力



ド呼出し操作<sup>8</sup>をおこなうと、図 3-a のように画面上に ASearch のウィンドウが現れ、入力フォーカスがメモ帳画面から ASearch 画面に移動します。

ASearch は、入力文字をキーワードとして動的曖昧検索を実行し、マッチした単語を画面下部に入力単語候補として表示します。たとえば「p」と入力すると、英単語辞書に対して動的検索が実行され、図 3-b のように「p」で始まる単語が一覧表示されます。

次に「i」を入力すると、画面が図 3-c のように変化し、「pi」で始まる単語が表示されます。

同様に「piteca」まで入力すると、自動的に曖昧検索が実行され、正しい綴りの単語「Pithecanthropus」が候補として表示されます(図 3-d)。

これを選択することによりこの単語がメモ帳に入力され、図 3-e のようにもとの画面に戻ります。

このように、動的検索システムを FEP として使うことにより、正しい綴りが分からない単語でも簡単に入力することができるようになります。この例の場合、ユー

ザーは「...nthropus」を入力する手間が省けたわけですから、入力の高速化も期待できます。

次回は、Pilot 上での ASearch の開発について詳しく解説します。

(ますい・としゆき ソニー CSL)

[参考文献]

- [1] Ben Shneiderman, Direct manipulation: A step beyond programming languages, *IEEE Computer*, Vol.16, No.8, pp.57-69, 1983
- [2] Ben Shneiderman, Dynamic queries for visual information seeking, *IEEE Software*, Vol.11, No.6, pp.70-77, November 1994
- [3] Sun Wu and Udi Manber, Agrep - a fast approximate pattern-matching tool, In *Proceedings of USENIX Technical Conference*, pp.153-162, San Francisco, January 1992
- [4] 増井俊之「富豪的プログラミング」、bit 1997 年 1 月号、pp.36-37
- [5] 増井俊之、水口 充、George Borden、柏木宏一「なめらかなユーザインタフェース」、第 37 回冬のプログラミングシンポジウム予稿集、pp.13-23、情報処理学会、1996 年 1 月

<sup>8</sup> Pilot の文字入力領域から表示画面中央に向けてペンをドラッグするが、あるいは文字入力領域の左下部をタップする操作。