グラフィクスとCAD 28 − 10
(1987・8・20・21)

Wnn*かな漢字変換システムにおける ユーザ・インタフェース

坂下 秀
(株)アステック

かな漢字変換システム Wnn のユーザ・インタフェースを報告する。その目標と特徴は、

- サーバ・クライアント・モデルを用いた文章一括変換可能なかな漢字変換システム
- 日本語入力フロント・エンドは強力な編集機能を持つ
- システムの全てがオープンであり、ユーザがシステムを自由にカスタマイズできる。

などである。

User Interface of Wnn Kana-to-Kanji Conversion System

Shiu Sakashita
ASTEC,Inc.

In this paper, I report on the User Interface of Wnn Kana-to-Kanji Conversion System. The fetures of Wnn are followings.

- Wnn, based on the client-server model, is the Kana-to-Kanji conversion system with customizable front-end user interface.
- Wnn offers a wide variety of dictionary utilities and editing functions.

*Wnn は、京都大学数理解析研究所、立石電機(株)、(株)アステックが共同で開発したかな漢字変換システムである。

1 はじめに

日本語の文書を作っていくときには、かな漢字変換システムのユーザ・インタフェースが、非常に重要である。しかし、現在の多くのシステムは、'使い易い'とは言えない。かな漢字変換のユーザ・インタフェースが悪いと、それを用いて書いた文章まで悪いものになってしまう。ここでは、このような問題点を改善したWnnかな漢字変換システムのユーザ・インタフェースを紹介する。

2 現在のかな漢字変換システム

現在のかな漢字変換システムには、次のような不満がある。

- 機能は高いがユーザ・インタフェースが悪い
  良い例は'逐次変換'である。これは、常に正しい変換を行うならば、非常に使い勝手が良いであろう。しかし、現在の技術水準では、誤変換が避けられないため、かなり変換結果を修正する必要がある。ところが多くのシステムは、その修正がかなり面倒である。
- ユーザ・インタフェースを変えることができない
  たとえば、ユーザは、
  - 変換行の編集キーのバインディング
  - ローマ字の綴り方
  - 変換のスタイル

  などを変えたいが、可能なものはほとんどない。
- その他
  - 辞書を自由に編集する
  - かな漢字変換を利用したアプリケーションを作る

  などもできない。

このように、多くのシステムでは、ユーザ・インタフェースが固定されている。しかし、日本語の入力は、個人によってさまざまなスタイルがあるはずである。それを、画一化してしまうことは、入力する文書まで画一化しかねない。

良い例としては、"ワープロで書くと、不必要に漢字を使ってしまう"ことがある。これは、ひらがなを漢字にしがちな'かな漢字変換システム'が原因ではないだろうか? これも、漢字に変換する比率を、ユーザが制御できれば、改善できるのではないだろうか?

3 Wnn

ここで紹介するWnnは、京都大学数理解析研究所が、アステック、立石電機の協力で開発したかな漢字変換システムである。

Wnnは、

- 開かれたシステム
- 計算機の機種に依存しない共通の日本語環境の整備

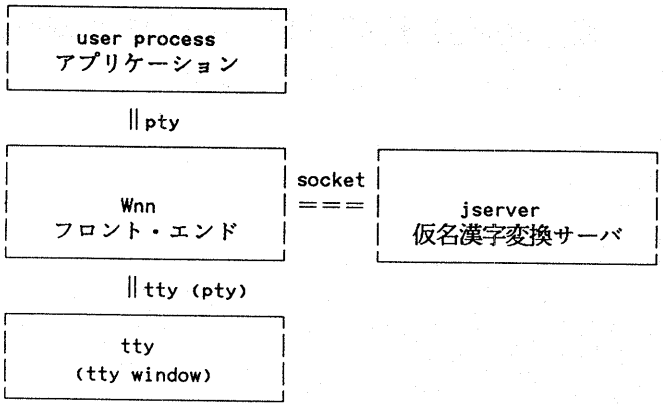

図 1: Wnn の構成

- ユーザ間でのユーティリティや辞書などの交換の促進

をめざしてつくられた。

Wnn の特徴は、

- UNIX[1]上で動作する、サーバ・クライアント・モデルを用いた文章一括変換可能なかな漢字変換システムである。
- 日本語入力フロント・エンドは強力な編集機能を持ち、円滑に日本語の入力を行うことができる。
- システムの全てがオープンであり、ユーザが、システムを自由にカスタマイズできる。

などである。

3.1 構成

Wnn は、サーバ・クライアント・モデルに基づいた構成になっている(図1参照)。

フロント・エンド(client)は、UNIX4.2bsd の socket 機能を使って、かな漢字変換サーバ(server)と通信を行い、かな漢字変換を行う。

フロント・エンドとユーザ・プロセスとは、pty(UNIX4.2bsd の仮想 tty 機能)で結んでいる。そのため、既存の tty ベースのアプリケーション(たとえば、vi,emacs などのスクリーン・エディタも含む)は、全く変更することなしに、Wnn の上で使うことができる。

4 Wnn のユーザ・インタフェース

4.1 n 文節仮名漢字変換

Wnn は、多くのかな漢字変換システムで採用している、'2 文節最長一致法'を一般化した、'後ろ向き n 文節評価値最大法'による文章一括変換を行う。

[1]UNIX は米国AT&T ベル研究所で開発されたオペレーティング・システムである。

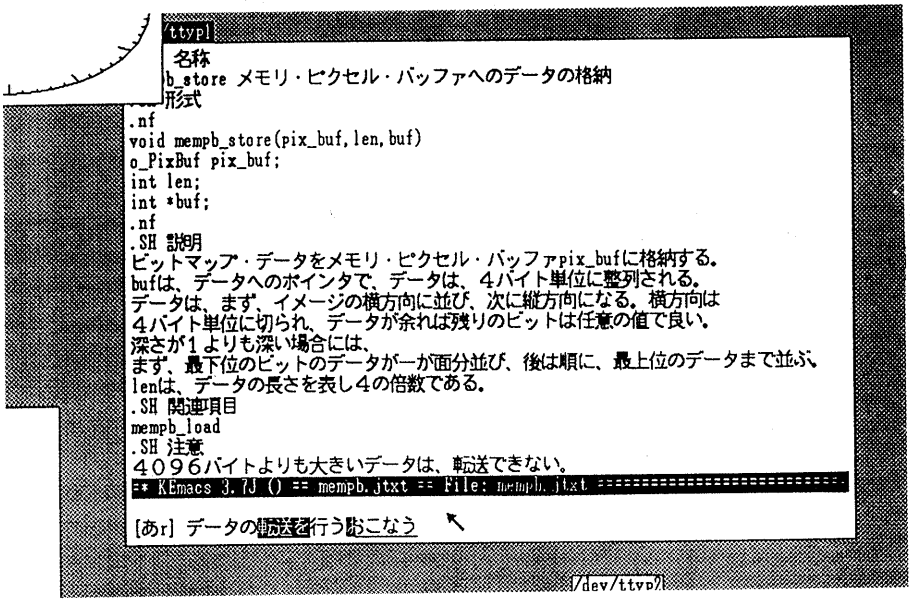

図 2: Wnn のフロント・エンド

評価値を決定する関数 f は 5 つのパラメータをとる。頻度 p_1、文節長さ p_2、自立語長さ p_3、今使ったよビット p_4、辞書プライオリティ p_5である。これらの重みつき和が評価値となる。

$$f(p_1, p_2, p_3, p_4, p_5) = ap_1 + bp_2 + cp_3 + dp_4 + ep_5$$

文節の数上の n の値や、重み a,b,c,d,e は、ユーザごとに指定することができる。これらのパラメータを変えることにより、さまざまな変換のスタイルを選ぶことができる。

たとえば、

$$a = c = d = 0, b > 0, n = 2$$

とすれば、2文節最長一致法になる。また、

$$a = 0, d > 0$$

とすれば、頻度更新は行わず、最後に使ったものを選ぶようにすることができる。

このように Wnn では、変換の基本アルゴリズムまでも、ユーザが容易にカスタマイズできる。

4.2 フロント・エンド

Wnn の標準のユーザ・インターフェースは、パーソナル・コンピュータでよくみられるような、フロント・エンドの形式である(図2参照)。このため、通常の漢字端末でも、ウィンドウ・システムの端末エミュレータのもとでも、同じフロント・エンドを用いて、同じユーザ・インターフェースで作業を行うことができる。

フロント・エンドは、 romkan(4.3参照)と呼ばれるローマ字漢字コード変換部と、マンマシン・インターフェイス部から成っている。

マンマシン・インターフェイス部は、

- tcsh[2]に似た編集/ヒストリ機能を持つ。

[2]tcsh は、UNIX4.2bsd のshell であるcsh の行編集とヒストリの操作の機能を強化したshell である。

```
(defvar (hex1 hex3) (list 2 3 4 5 6 7))
(defvar (hex2 hex4) (list 0 1 2 3 4 5 6 7 8 9 A B C D E F))
X20(hex3)(hex4) (error)
X7F(hex3)(hex4) (error)
X(hex1)(hex2)20 (error)
X(hex1)(hex2)7F (error)
X(hex1)(hex2)(hex3)(hex4) (+ (+ (* (value (hex1)) '\x1000')
                               (+ (* (value (hex2)) '\x100')
                                  (+ (* (value (hex3)) '\x10')
                                     (value (hex4)))))
                            '\x8080')
```

図 3: romkan による JIS コード変換

- 単語登録や使用辞書の選択を、フロント・エンドから簡単に行うことができる。
- フロント・エンドの各コマンドのキー・バインドは、すべてユーザ・プログラマブルである。

などの特徴がある。

これらの機能を用いると、Wnn の上から起動する tty ベースのアプリケーションのすべての行入力で、あたかも、編集/ヒストリ機能がついているかのような操作を行うことができる。

4.3 romkan(ローマ字漢字コード変換)

romkan は、ローマ字で入力された文字列を、変換規則にしたがって、漢字コード列に変換する。変換規則は有限状態オートマトンのテーブルになっており、ユーザが自由にプログラムすることができる。

romkan は、単純な'かなローマ字変換'ばかりではなく、'kk'という入力に対して、'株式会社'に変換することができる。通常のかな漢字変換システムでは、辞書に'株式会社'の読みとして'kk'を登録しなければ、このような変換はできない。

さらに、図3の romkan のプログラムを使うと、JIS コード変換を行うことができる。'X2144'と入力すれば、'…'に変換される。

意味は、

```
X(hex1)(hex2)20(error)
```

などと書いてあることによって、'X2120'といった、JIS コードの範囲外の入力には、エラーを警告することができる。

```
(value(hex1))
```

は、変数 hex1 の値が 1 という数字のとき、1 というコードを返す。

```
\x1000
```

は、16 進の 1000 という数の記述で、単に 1000 と書くと文字列と見なされてしまうためである。

最後に、16 進の 8080 を足しているのは、romkan が文字の表現に、内部 EUC 2 バイトコードを使っているためである。

4.4 辞書

Wnn では、ユーザが自由に辞書を編集できるようになっている。

ユーザは、辞書のマージとソートを行うことができる。したがって、他のユーザの学習結果を容易に取り込むことができる。また、システム形式の辞書をアスキー形式に変換することにより、辞書を自由に変更できる。アスキー形式の辞書をシステム形式に変換し、ユーザ独自の辞書を登録することもできる。

さらに、このように編集したユーザの辞書は、フロント・エンドによって、動的に Wnn に組み込んだり外したりできる。

4.5 アプリケーション

Wnn はユーザに、'romkan'と'jlib'の 2 つのライブラリを提供している。

romkan は、4.3で説明したローマ字漢字コード変換機能を、ライブラリにしたものである。jlib は、フロント・エンドとかな漢字変換サーバとの通信機能を中心に、変換機能と辞書操作などをライブラリにしたものである。これらのライブラリを用いることにより、容易に Wnn を利用したアプリケーションを作成することができる。

5 まとめ

以上のように、Wnn は開かれたシステムになっている。

この Wnn を使ってみてわかったことは、

- かな漢字変換の効率を高めることは重要だが、それ以上に、ユーザ・インタフェースを整備するほうが重要である。
- ツールをつくるようになった。
  Wnn は、開かれたシステムなので、ユーザが自由にシステムをカスタマイズできる。そのため、Wnn を使いやすくするツールがいくつかできている。たとえば、さきほどの romkan を利用した JIS コード入力などがある。

といったことである。

しかし、問題点もある。それは、そのカスタマイズがかなり面倒なことである。その原因は、

- カスタマイズ用のファイルの記述が難しい。とくに、romkan の記述は難しい。
- ドキュメントが整備されていないため、カスタマイズの方法がわからない。
- バージョンが一定しないため、カスタマイズの方法がすぐ変更される。

などがある。

したがって、開かれているとはいえ、ユーザ全員にとって開かれているわけではない。つまり、その気になれば原理的にはどのようなことでもできるが、カスタマイズをするインタフェースが非常に悪いため、カスタマイズが困難である。

Wnn は、まだまだ開発途上であるがゆえの欠点も多いが、今後は、

- ツールの整備。
- ドキュメントの整備。

- より良いユーザ・インタフェースの整備。たとえば、ウィンドウ・システムを生かしたかな漢字変換インタフェースの開発。
- クライアント・サーバ・モデルを生かして、ネットワークに対応するための Wnn の拡張。

を行っていきたい。

6 謝辞

本稿を用意するにあたっては、Wnn の開発者のみなさん、特に、京都大学数理解析研究所の新出尚之氏、立木秀樹氏、(株)アステックの竹岡尚三氏との、JUNET の mail/news での議論が非常に参考になった。ここに感謝する。

References

[1] 萩谷昌己: GMW ウィンドウ・システムについて, bit, Vol.19, No.3, pp.226–241(Mar 1987)

[2] 桜川貴司: 開かれた日本語入力システム Wnn, bit, Vol.19, no.10, pp.13–23(Sep 1987)

[3] 竹岡尚三 他: Wnn 日本語処理システム, *jus 9th UNIX SYMPOSIUM PROCEEDINGS*, pp. 21–34(Jul 1987)

討論

１０．かな漢字変換におけるユーザインタフェース　　坂下（アステック）

鎌田：サーバのアクセスする辞書はユニバーサルなものですか。

坂下：２つの辞書をアクセスします。世界に１つシステム辞書があり、クライアントごとにユーザ辞書があります。

鎌田：そうすると、学習した結果はユーザ辞書の方に書き込むわけですね。
学習に関してはどのようなデータのため方をしているのでしょうか。

坂下：基本的には頻度でやっています。文節の切り方の学習は難しくてまだ完全ではありません。

鎌田：連文節で文節の評価をする場合、文節の長さはどの文節の長さで評価するのですか。また、特別な助詞がついたら切るようにしたいことがよくあるのですがどうでしょうか。

坂下：文節の長さは後ろから見ています。助詞の接続と同じようなことで接頭語、接尾語の処理がありますが、これらの接続の仕方はユーザに解放することで逃げています。

守屋：辞書は自作のものですか。

坂下：電子協のものを使っています。ただし、ＣＰＵごとにロイヤリティをとられるので、当社では複数システムをネットワークでつないで辞書を１つで済ませています。

林：　そのときサーバはネットワークを通してユーザ辞書を読みに行くのですか。

坂下：辞書は最初に全部読んでしまうのでディスクをアクセスしません。

林：　それはかなりひんしゅくを買いませんか。

坂下：昔は買いましたが、今は大丈夫です。現在の環境はＳｕｎ－３の２６０に５００メガのディスクがついています。

守屋：機能のキーマップの定義はどのようにするのですか。

坂下：機能名の後にキーの列を書いて定義します。

守屋：ローマ字からの変換のところはどうなっていますか。

坂下：入力列を認識するオートマトンになっていて、計算もできます。したがって電卓を作ることができます。