

5 Q-7

## 異種環境の統一的な日本語化 — Wnnを用いたCedar環境の日本語化 —

前田 正浩、倉橋 政之、守屋 康正

(富士ゼロックス株式会社 システム技術研究所)

### 1. 概要

サーバ/クライアント方式に基づいたかな漢字変換機能の実装により、異なるプログラミング環境で共通に利用できる日本語入力機能の有効性を確認したので報告する。

### 2. 背景

Cedar言語/環境[1]はXeroxコーポレーションのパロアルトリサーチセンターにおいて開発された実験的なプログラミング言語/環境である。Cedar環境はライトウェイトプロセス・単一アドレス空間などを特徴とし、UNIX<sup>\*1)</sup>とは大きく異なる環境である。

研究環境としてUNIX/Cedarなどの環境を使用している当研究所では、統一された日本語入力機能の実現を目的として、サーバ/クライアント方式に基づくかな漢字変換であるWnn[2]<sup>\*2)</sup>に注目し、Cedar環境にWnnクライアントを試験的に実装することにより、UNIX/Cedar両環境において同一の日本語入力環境を構築することにした。

\*1) UNIXオペレーティングシステムはAT&T社ベル研究所が開発したソフトウェアであり、AT&T社がライセンスしている。

\*2) Wnnの著作権は京都大学数理解析研究所・OMRON・ASTECに帰属する。

### 3. 日本語入力環境の共有化

従来、日本語入力機能は個々の環境上で全てを実装し、利用される方式が主流であったが、サーバ/クライアント方式に基づく方式では

- ・ 辞書の共有化(個人およびグループ内)
- ・ ディスクスペースファクタの経済化

といったメリットを得ることができる。更に複数異種環境でサーバ/クライアント方式に基づく日本語入力機能を共有することにより

- ・ 複数異機種を所有するユーザに対する同一日本語入力機能の提供

- ・ 複数異機種環境での日本語入力機能実現コストの極少化

といったメリットを得ることができる。

### 4. 実装

#### 4.1 方針

- ・ それぞれの環境での日本語入力における機能とキーとのアサインを統一する[3]。
- ・ 日本語入力機能と既存のアプリケーションとを共存させ、日本語化によって使用できなくなるアプリケーションを発生させない。

#### 4.2 実現手段の検討

UNIX及びCedarの両プログラミング環境で共有できる日本語入力機能を実現するため、

- ・ サーバ/クライアント方式
- ・ 仕様が公開されている
- ・ UNIX環境で広く普及している

等の条件を満たすWnnを今回の日本語変換サーバとして採用した。

WnnをUNIX環境と同等のレベルでCedar環境上で実現する為には、サーバ/クライアント間における通信ライブラリを必要とする。本部分はWnnの通信ライブラリを参考にCedar言語で構築した。

Cedar環境における標準的なテキストエディタはTioga[1]と呼ばれるエディタであるが、このエディタとかな漢字変換パッケージとの接続方法が確立すればCedar環境上での日本語編集が可能となる。今回Cedar環境に影響を及ぼす実装法を避けるため、エディタを書き直すことはせずに、エディタで提供されている外部ルーチン用インターフェースを利用してフロントエンドプロセッサを組み込んだ。

#### 4.3 構成

##### 4.3.1 システム構成

図1にWnnが動作するためのシステム構成図を示す。今回実装した日本語入力機能(パッケージ

Integrated Japanization method for different programming environments.

Masahiro MAEDA, Masayuki KURAHASHI, Yasumasa MORIYA

System Technology Research Laboratory, Fuji Xerox Co., Ltd.

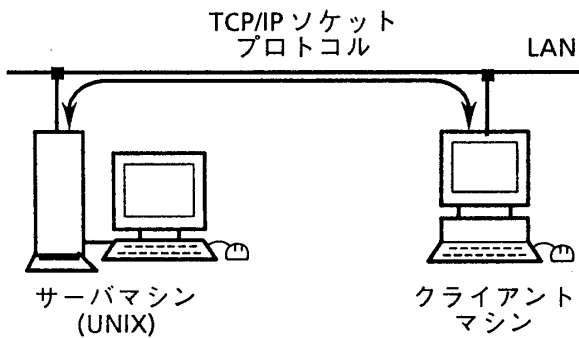


図1 システム構成

名:RKanji)は、この環境で動作する。

Wnnの主要機能である かな漢字変換プログラム (jserver)はサーバとしてUNIX環境上で作動する。クライアント側(Cedar環境)で動作するRKanjiは、UNIX環境上のjserverとLANを通じてソケット通信を行い、ひらがなと漢字の文字情報をやり取りすることにより、Wnnのかな漢字変換サーバ(jserver)のクライアントとして動作する。

#### 4.3.2 ソフトウェア構成

今回実装したRKanjiは次の三つの部分に分割される。

1. ローマ字かな変換部
2. jserver インタフェース (jserver I/F) 部
3. エディタインタフェース(エディタI/F) 部

図2にRKanjiと他のパッケージとの関連を示す。

TIP(Terminal Input Processor) はユーザのマウス/キー入力をユーザが定義したデータに変換し、エディタI/F部に通知する。次にローマ字かな変換部で漢字の読みを表すひらがなに変換され、jserver I/F部により、jserverに送られる。jserverでかな漢字変換された結果が、再びjserver I/F部に送り返され、エディタI/F部からエディタの表示ルーチンをコールすることにより、漢字文字列としてディスプレイ上

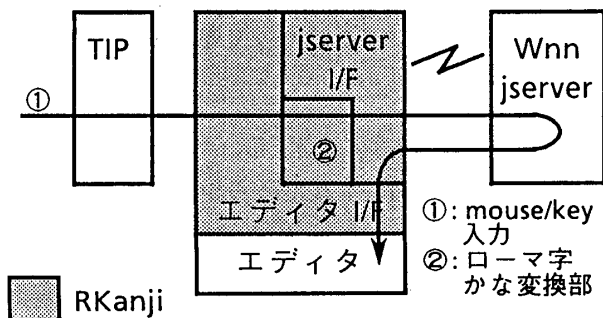


図2 ソフトウェア構成

に表示される。

#### 4.3.2.1 ローマ字かな変換部

ローマ字かな変換部は、ローマ字で入力された漢字の読みをひらがなに変換する。

#### 4.3.2.2 jserver I/F部

jserverは、サーバとクライアントの通信をTCP/IPのソケット通信で行っている。Wnnバージョン4(Wnn V4)では、この部分の機能をC言語で記述された通信ライブラリ(jslib)として提供している。RKanjiではjslibに相当する機能をCedar言語で実装することにより、jserver I/F部の機能をCedar環境上で実現している。

#### 4.3.2.3 エディタI/F部

この部分は、RKanjiにおいて、TIPからのマウス/キー入力通知の窓口の働きとRKanjiの処理に必要なエディタのルーチンをコールする働きの二つを行う。RKanjiの処理に必要なエディタのルーチンの例としては、エディタウィンドウに表示された文書に漢字を入力するルーチンなどが挙げられる。

### 5. 結果

1. 異種アーキテクチャ/異種OS/異種環境であっても、Wnn用クライアントは、ホスト名-アドレス変換と、TCP/IPのソケット通信機能を持つプログラミング環境であれば、実装可能であることが確認できた。
2. 今回の作業量は実質3人月であった。結果的に異機種上において日本語入力機能を実現/開発するにあたって本方式が省実装コスト性・移植の容易性が良いことが確認された。

### 6. まとめ

サーバ/クライアント方式は、処理の分散・実装の容易さ・リソースの有効利用、などといった点で有利であることは周知の事実であるが、今回更に異種環境の結合/統合にも有効であることが確認できた。

### 7. 参考文献

- [1] W. Teitelman: "A Tour Through Cedar", IEEE software, Vol.1, No.2, pp44-73
- [2] Wnnオンラインマニュアル
- [3] 戸村,他: "ネットワーク仮名漢字変換サーバを用いた日本語環境:「たまご」", 情報処理学会研究報告 89-SE-64-22