字形データの共有と情報交換が可能な動的拡張可能文字集合 DCS

江畠隆行、板野肯三、新城靖
筑波大学大学院システム情報工学研究科

1 はじめに

現在もっとも多く利用されている漢字の文字集合規格は JIS X 0208:1997 である。これには 6879 文字が含まれているが、人名や地名なので不足が目立つようになった。そこで JIS X 0213:2000 において 11223 文字に拡張された [1]。規格を拡張する事によりコンピュータで扱える文字を増やす方法には次の 2 つの問題がある。

1. 規格の改訂に時間がかかる。
2. 規格と実装の乖離が起きる [2]。規格が必ずコンピュータに実装されるとは限らない。

このような問題に対処するために、今までは外字が使われてきた。しかし外字には情報交換には使えなくなってしまうという問題がある。本研究では、動的拡張可能文字集合 DCS(Dynamically extensible Character Set) でこれらの問題を解決する。

2 動的拡張可能文字集合 DCS

2.1 DCS における文字の概念

通常の文字集合では、その文字を使う利用者のグループの中で、文字のアイデンティティ (その文字がどういう文字であるかということ) は、先に合意されている。これに対して、DCS では文字のアイデンティティに対する合意を求めない。その代わり文字を定義するために字形データを用いる。字形データとは、例えば 16 × 16 ドットのビットマップで文字を表したデータである。DCS では、各利用者は字形データを用意することによって新しい文字を定義することができる。これで規格の改訂を待つことなく、新しい文字を利用できるようになる。また、文字を定義した時には必ず最低 1 個の字形データが存在するので規格と実装の乖離問題が解消される。

DCS は初期値として主要文字集合を含める。具体的には Unicode, JIS(日本), GB(中国), KS(韓国), TCVN(ベトナム) で規定されている文字集合の字形データを登録する。したがって、DCS をサポートしたシステムではこれらの文字を表示することができる。

2.2 DCS コードとローカルコード

DCS コードとは、DCS で定義された文字を指定するための 32bit の長さを持つ文字コードである。

既存の多くのアプリケーションはその OS が用いる文字コード (多くは 16bit 長さ) に対応している。よって、既存のアプリケーションでは DCS コードで表現されたテキストを処理できない。この問題を解決するために、ローカルコードという概念を導入する。

ローカルコードとは、その OS が標準で利用しているコードである。ローカルコードに DCS コードの一部を仮想記憶と同様に動的にマップすることで、DCS コードのテキストを扱う。図 1 はローカルコードとして JIS を用いているシステムでのマッピングの例を示している。DCS コードの JIS の領域は、対応するローカルコードにマップされている。その他に、Unicode で定義されている文字やユーザが定義した新しい文字が、ローカルコードの JIS で未使用の部分にマップされ利用可能になっている。未使用の領域がなくなった場合、仮想記憶と同様に当分利用していない文字を犠牲にして必要な新しい文字を上書きする。

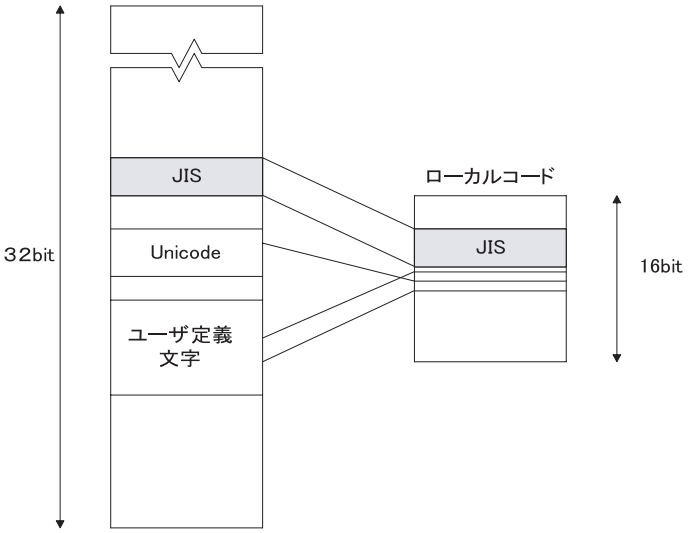

図 1: ローカルコードへのマッピング

3 DCS を実現するサーバとユーティリティ

図 2 に DCS を実現するためのサーバとユーティリティーを示す。この図は 2 つのホスト間でテキストを交換している様子を示している。

DCS サーバは内部にハッシュ表を持っている。このハッシュ表は DCS コードをキーとし、字形および読みを値とする。

送信側および受信側ホストには、ローカルコードと DCS コードの対応表がある。これは 2 つのハッシュ表からなり、現在利用されている DCS の文字の DCS コードからローカルコードへ、または、ローカルコードから DCS コードへ高速に変換することができる。

送信側で新しい文字を定義したい利用者は、まず FontEditor を用いて新しい字形を作成し、それを読みとともに DCS サーバに送る。DCS サーバは、新しい字形を受け取ると、新たに DCS コードを発行して FontEditor に返す。

FontEditor は、新しい文字の DCS コードを受け取ると、ローカルコードの未使用部分を探し、その DCS コードとローカルコードの対応を登録する。FontEditor は、そのローカルコードを用いてかな漢字変換サーバの辞書に対して

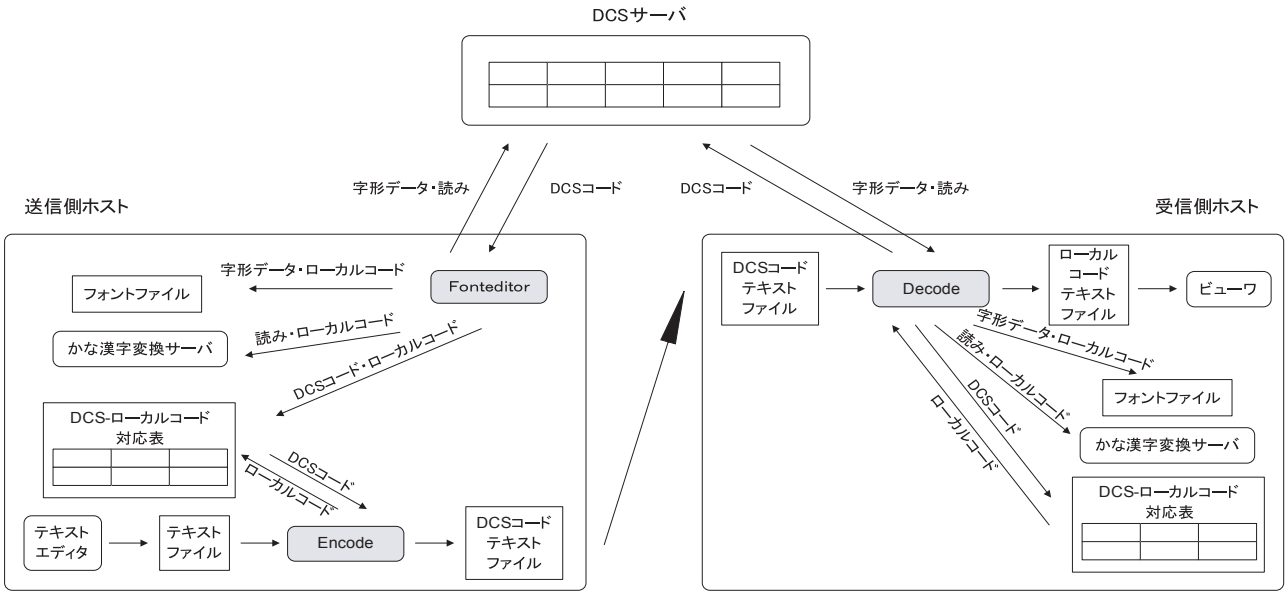

図 2: DCS サーバと送信側ホスト

読みを登録する。FontEditor は最後にローカルコードを用いて、字形データをウインドウシステムのフォントファイルに書き込む。

利用者は、通常のテキストエディタを用いて新しい文字を含んだテキストを作成することができる。この時、既存の文字と同じように、かな漢字変換により入力することができる。利用者はテキストエディタで、ローカルコードでテキストを保存する。新しい文字を含むテキスト（ローカルコード）を他のホストに送信する時には、Encode ユーティリティにより、DCS コードのテキストに変換する。

新しい文字を含むテキストを受信した利用者は、Decode ユーティリティを使用して、そのホストのローカルコードに変換する。Decode ユーティリティは、対応表を参照しながら DCS コードからローカルコードへ変換する。対応表にその DCS コードの文字が含まれていない場合には、 DCS サーバに DCS コードを送り、DCS サーバから字形データを受け取る。そして、ローカルコードの未使用の部分を探し、その DCS コードとローカルコードを対応を登録する。最後に、字形データをウインドウ・システムのフォントファイルに書き込む。

DCS サーバを Java で記述し Glue[3] を使用して、XML Web サービスのサーバとして実装した。ハッシュ表を Berkeley DB[4] で実装した。

DCS クライアントと通信を行うためのインターフェースを以下に示す。

字形データと読みを受けて、DCS **コードを返す**

```
put(BdfElement data, String yomi);
```

DCS **コードを受けて、字体データを返す**

```
BdfElement get(DcsCode code);
```

DCS **コードを受けて、読みを返す**

```
String getYomi(DcsCode code);
```

ここで BdfElement とは、X11 Window System で用いられる BDF 形式の字形データを意味する。

4 関連研究

大澤らは、プログラム可能文字コードシステム EPICS を提案している [5]。これはプログラム可能な仮想マシンを使用して Unicode とともに利用者定義文字を扱える。

EPICS では利用者定義文字を管理するサーバがなく、利用者は広大な可変長のコード空間の中で互いに重ならないように拡張していく。これに対して DCS では、文字コードは固定長で DCS サーバにより大域的に一意に定まる。また利用者間で字形データを共有する事が容易である。

5 まとめと予定

動的拡張可能文字集合 DCS を提案した。そして、実現のための DCS サーバとユーティリティの開発を行った。今後の課題は DCS サーバに蓄積されている字形データをブラウズする機能を実装することである。

参考文献

[1] 芝野 耕司:“JIS 漢字字典”, 日本規格協会,2002.

[2] 川俣明:“パソコンにおける日本語処理/文字コードハンドブック” 技術評論社, 1999.

[3] webMethods: “GLUE 5.0.2”, 2004 http://www.webmethods.co.jp/dev/glue.html

[4] Sleepycat:“Berkeley DB”, New Riders Publishing, 2001.

[5] Noritake OSAWA and Toshitsugu YUBA: “An Efficient, Programmable and Interchangeable Code System: EPICS”, IEICE Trans. on Information and Systems, Vol.E83-D, No.4, pp.797-806, 2000.