マルチリンガルエディタにおける検索機能の設計(2)

1ZA－7

検索機能の実現

上園一知　片岡朋子　筧捷彦†

早稲田大学メディアネットワークセンター
†早稲田大学理工学部

1. はじめに

任意に複数文字種を混在した文書が編集可能な、マルチリンガルエディタにおける検索機能の実装レベルとして、「基本検索」と「発展型検索」に分類することができた[1]。これは、文字がもつ情報を「文字属性」と「フォント属性」に分離し、それぞれの組み合わせにより実現することが可能である。

そこで、検索機能を実装するために必要な情報を分析し、設計を行った。この際、検索文字列と被検索文字列がメモリ上でずれを生じる。本稿では、エディタの実現上起こり得る文字列のずれの問題について触れ、各機能に必要な情報と実装方法について述べる。

2. 文字列検索の実装で生じる問題

任意に複数の文字種が混在した文字列では、文字の表記方向が混在するため、文字の表示上の配置順序は、メモリ上の順序と異なる。

よって、文字列検索を、メモリ上の順序に準じた検索と、表示上の順序に準じた検索に分離し、表示上の配置順序とメモリ上の順序のずれを考慮する必要がある[2]。

ずれを考慮して、検索文字列と被検索文字列をそれぞれ表記方向ごとにブロックとし、ブロックごとに検索する。検索文字列のブロックの境界では表記方向が変化するので、マッチした被検索文字列の末端の文字が、ブロックの末端かどうか検査する。

Designing Search Functions for a Multilingual Editor (2): Realization of Search Functions.
Kazutomo Uezono, Tomoko Kataoka and Katsuhiko Kakehi
Waseda University
{uezono, tomoko, kakehi}@kake.info.waseda.ac.jp

また、文字列は図形配置開始位置に従い表示されるので、検索文字列と被検索文字列の図形配置開始位置について検査する(図 1)。

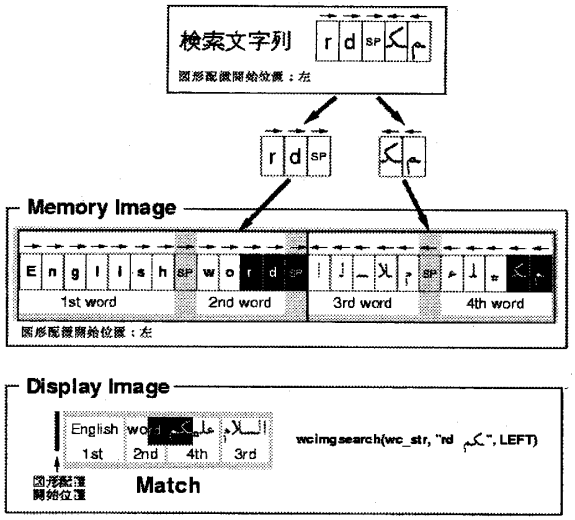

図 1. 表記方向が混在した文字列の検索

以上より、表示上の配置順序に従った文字列検索が可能である。

3. 基本検索機能の実現

任意に複数文字種が混在した文書で、すべての「文字」を検索するためには、それぞれの文字が統一された識別子をもつ必要がある。さらに、文字は識別子以外にも情報をもつ。

文字の分析により、一般化した文字を定義し、文字のもつ情報を導出することができた(表 1)[3]。文字コード単位(mb)ではこれらの情報が含まれていないため、これを単位とした文字列処理は煩雑となる。そこで、これらの情報を含んだ固定長の文字列処理単位として、WC(Wide Character)を国際化ライブラリ[3]で実装した。

表 1. 文字の主な属性情報

識別情報	所属情報	表示情報
文字 ID	文字集合 ID	図形配置開始位置
	文字集合への変換情報	改行方向
		行配置開始位置
		禁則処理情報
		Direction
		Form
		Variant
		Ligature

基本検索は、文字の識別子(文字 ID)に基づいた検索である。マルチリンガルエディタは、国際化ライブラリを用いて開発するので、このエディタで処理する文字列は、WC で保持される。そのため、WC を単位とし、文字 ID のみを参照して検索を行う。

4. 発展型検索機能

4.1. 特定の表示図形の検索

文字のフォーム(Form)・異体字(Variant)・リガチャ(Ligature)を検索する場合、表示図形を特定する必要があるので、文字 ID のほか文字属性の表示図形情報(Form・Variant・Ligature)を利用する。

4.2. Discontinuous Sequence 検索

Diacritic Mark を無視した検索を実現するために、Diacritic Mark の扱いを定義する必要がある。

Diacritic Mark を基本文字に併記した場合、扱いとして、1)単独文字、2)元文字の異体字、と定義することが可能である。しかし、このような検索が必要な文字群では、Diacritic Mark が併記された文字を別字として判断するのではなく、同一の文字として判断するので、2)として定義するべきである。

このとき、Diacritic Mark は WC の一情報であり、異体字情報として実現される。よって、この検索では文字 ID のほかに、WC の異体字情報を利用する。

4.3. 同図形異文字検索

異なる文字が同一の表示図形で表現される場合、これらの図形表現の同一性は、WC のもつ表示情報を利用しても、文字が異なるので、判断することができない。すなわち、図形表現の判断には、フォント属性を使用する必要がある。

フォントは、文字列の表示時にインタラクティブに指定される。そこで、WC からフォントに変換する際に、フォント属性をもたせたフォント情報列(Font Information String)を作成し、フォント情報列がもつ識別子を参照して検索する。

4.4. フォントの書体に着目した検索

文字の表示に異なる書体を使用して、強調などの修飾を行ったとき、その検索方法として、単に修飾表現を検索する場合と、特定文字列の修飾表現を検索する場合に分離することができる。

前者の場合、修飾表現で使用しているフォントを指定すればよいので、フォント属性を利用する。すなわち、フォント情報列のみ参照すれば、実現可能である。

後者の場合、更に文字列を指定しているので、フォント属性に加え、文字 ID などの文字属性を利用する。このとき、WC とフォント情報列を同時に参照し、検索する必要がある。

5. まとめ

任意の複数文字種が混在した文書では、文字の使用方法や表示方法が複数存在するため、検索機能を分類する必要がある。そこで、検索を「基本検索」「発展型検索」に分類し、それぞれに必要な機能および実現に必要となる情報について述べた。

文字属性とフォント属性とを峻別し、それを組み合わせることによって実現可能であり、マルチリンガルエディタの検索機能として実装予定である。

参考文献

[1] 片岡朋子他: マルチリンガルエディタにおける検索機能の設計(1): 基本要件の抽出と発展型の提案, 第 59 回情報処理学会全国大会講演論文集(4), 1999.

[2] 上園一知他: 国際化 Web Browser の設計, 情処研報 Vol.99, No.62, pp. 57-64, 1999.

[3] Uezono, K., et al., The Internationalized Environment Enabling Proper Text Processing, 第 55 回情報処理学会全国大会講演論文集(4), pp. 47-48, 1997.