

国際化 Web Browser の設計

上園 一知 片岡 朋子 笈 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

〒 169-8555 東京都新宿区大久保 3-4-1

{uezono, tomoko, kakehi}@kake.info.waseda.ac.jp

あらまし

インターネットなどの情報通信で使用できる文字はラテンアルファベットが主流であり、ネットワークの国際化の反面、未だ母語を利用できず、ソフトウェアの国際化は進んでいないのが現状である。World Wide Web においても、HTML 自身の仕様と Web Browser が特定のエンコーディングを対象としているため、任意に文字を混在させて利用可能な国際化の実現が困難である。そこで、X Window System 上で開発した国際化ライブラリである System 1 を利用し、国際化 Web Browser の開発を行っている。本稿では、HTML 文書の表示と文字列検索を考察し、Web Browser に必要な機能について述べる。

キーワード 国際化、テキスト処理、文字列表示、文字列検索、HTML、Web Browser

The Design of Internationalized Web Browser

Kazutomo Uezono Tomoko kataoka Katsuhiko Kakehi[†]

Media Network Center, Waseda University

[†]School of Science and Engineering, Waseda University

3-4-1 Okubo, Shinjuku-ku, Tokyo 169-8555 Japan

{uezono, tomoko, kakehi}@kake.info.waseda.ac.jp

Abstract

Even in such a highly cross-communicated society, the almost only script available in any computing network is still Latin alphabet, for which many native users cannot make full use of their own scripts. In the WWW environment as well, the true internationalization with any number of script mixed is hard to realize due to the specification of HTML itself and the specific encoding(s) of a Web Browser. To realize true internationalized computing, Internationalized Web Browser has been developing using internationalized library called System 1, which has already developed on X window system. This paper describes the methods and functions to display and search character strings in HTML documents on the Web Browser.

key words Internationalization, Text Processing, Text Displaying, Text Search, HTML, Web Browser

1 はじめに

インターネットの普及により、電子メールや World Wide Web (WWW) に代表される、国際的な情報通信が日常的に行われている。これらの情報は、テキストが大部分を占めており、ラテンアルファベットや漢字などの言語・文字が使用されている。

しかし、特にラテンアルファベットを使用しない言語を母語とする地域では、コンピュータでの母語の利用は未だ困難である。ハードウェア (物理的ネットワーク) が国際化したにもかかわらず、ソフトウェア (OS やアプリケーション) が未だ国際化されていない。これでは、コンピュータネットワークが完全に「国際化」したとは言いがたい。

現在までに、英語と英語以外の母語を同時に利用するために、各国・各地域で文字コードやエンコーディングを規定し、いくつかの言語・文字を混在したテキスト通信が利用されているが、すべての言語・文字の混在処理を可能とした、本質的な国際化は行われていない。

近年急速に利用が普及した WWW においても、公開されている HTML[1] 文書は既定のエンコーディングに基づき作成されているため、任意の文字を組み合わせることができず、画像を利用する例が多い。この場合、文字データとしては扱われないため、表示においても他の文字フォントとの統一がとれず、また文字列の検索をすることができない。これは HTML の仕様自体の問題と、ユーザインタフェースである Web Browser の機能の問題に起因していると考えられる。

そこで、任意に文字を混在可能な Web Browser の開発を行っている [2]。現在までに、すべての文字コード・文字を任意に混在可能な国際化ライブラリ (System 1) を、X Window System 上で実装し、既にほとんどの機能の実装を終了している (図 1)[3][4]。Web Browser は、このライブラリを利用して作成する。

本稿では、Web Browser の重要な機能である、

- 文字列の表示
- 文字列の検索

の実現手法について考察を行う。国際化テキスト処理の基礎となる「文字」、「表示図形」と「文字コード」の関係を定義する。その上で、「表示図形」の画面配置手法、「文字」の持つ情報を利用した文字列検索手法について、特に HTML 文書を対象としてそれぞれ考察し、Web Browser に必要な機能を導出する。

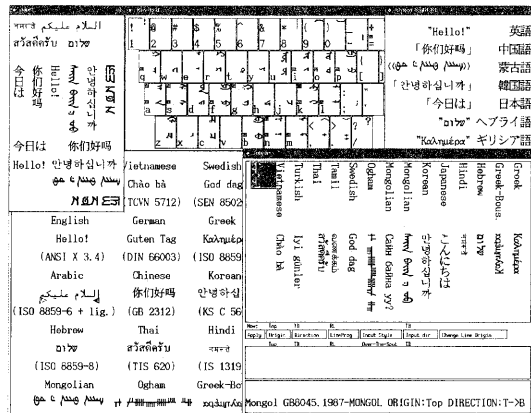


図 1: System 1 を使ったアプリケーション例

2 文字コード・文字・図形の関係

2.1 文字コードとエンコーディング

文字をコンピュータで利用するために、「情報交換用符号」として国際規格・国家規格などで規定されている。規定では、「制御文字集合」や「図形文字集合」が定義され、さらにビット並びを割り当てた符号化文字集合が定義される。ISO 2022[5] などによる符号拡張を行う場合、ビット並びは再割り当てが必要となる。本稿では、文字コードを「制御文字集合」および「図形文字集合」の総称、エンコーディングを「制御文字集合または図形文字集合へのビット並び割り当て方法」と定義する。

文字コードの利用には言語情報・地域情報が必要であると考えられている。特定の文字コードに対しエンコーディングを定め、言語情報・地域情報・エンコーディングを Locale として定義し、アプリケーションがこれを利用する形態 (POSIX Locale Model[7]) が多い。IANA[8] に登録されているエンコーディングも同様である。

この場合、Locale を複数用意しても、アプリケーションは一時に 1 Locale しか使用可能でないため、すべての言語・文字・文字コードを包含することはできない。そこで、エンコーディングの排他性を分析するために、文字コードとエンコーディングの関係に着目し、文字コード・エンコーディングの分類を行った (表 1)。

ISO 2022 は、文字コードの識別子としてエスケープシーケンスを規定しており、ISO 2022 で定められている文字コードの形式に従っていれば、文字コードをインタラクティブに切り替えて混在使用することが可能である。

ISO 10646-1[6] は ISO 2022 のエスケープシーケンスを使用せず、予め領域を設定し文字を割り当てて、複数の言語・文字を使用しようとするものである。ISO 10646-1

表 1: 文字コード・エンコーディングの分類

分類	エンコーディング	文字コード
ISO 2022 準拠型	ISO8859-1, ISO-2022-JP, EUC-JP, GB2312, EUC-KR など	ISO 8859-*, ANSI X 3.4, JIS X 0201, JIS X 0208, GB 2312, KS C 5601 など
ISO 10646-1 型	UTF-8, UTF-16, UCS-2, UCS-4 など	ISO 10646-1, Unicode
Shift_JIS 型	Shift_JIS, BIG5, VIQR, Windows-* など	BIG5, JIS X 0208, VIQR, Microsoft Windows Codepage など

に関しては、ISO 2022 から変更するエスケープシーケンスと、ISO 2022 への復帰エスケープシーケンスが規定されており、ISO 2022 との同時使用は可能である。

Shift_JIS 型エンコーディングの形式は対象とする文字コードごとに異なる。ISO 2022 と Shift_JIS 型エンコーディング間に何の規定もないため ISO 2022 と同時に扱うことはできないが、ISO 10646-1 と同様なエスケープシーケンスを定義できれば混在使用が可能となる。

以上より、ISO 2022 を基本とし、他のエンコーディングに対し識別子を定義すれば、すべてを混在使用することが可能である。また、ISO 2022 のデフォルト状態としてエンコーディングを与えることで、既存の HTML 文書との互換性は保持可能である。

2.2 文字コードと文字の関係

アラビア文字では、文字図形が隣接した文字との結合性により変化し複数存在するが、ISO 8859-6 ではすべての図形が定義されていない。この結合性はアラビア語などの言語に固有であり、テキスト処理は言語に依存していると考えられている。しかし、テキスト処理の単位である文字・表示単位である図形を言語非依存に定義が可能であれば、テキスト処理は Locale に依存しないので、任意に複数文字種を混在して処理することが可能である。

2.2.1 文字の定義

文字は、一般に文字が示す音価や意味により分類されているが、文字のもつ音価や意味は言語ごとに異なることから、言語に依存した分類であり、「文字」の本質的な分類とは言えない。複数の音価・意味を指示する記号として、文字は存在すると考えることができる。

そこで、文字の構造を基準とした分類を図 2 に示す。

文字は構造をもつ文字ともたない文字に分類することができる。もたない文字は、記号構成ルールが空集合であるとみなせば、

$$[\text{文字}] = [\text{記号}]^+ + [\text{記号構成ルール}]$$

$$[]^+ \text{は 1 回以上の繰り返し}$$

と一般化して定義することができる。

2.2.2 文字コードのコードポイントとの写像関係

ISO 8859-1 と JIS X 0201 では、ラテンアルファベット 'A' が定義されているが、コンピュータ上では異なる文字として認識される。よって、コンピュータ上の文字は言語により定義されるのではなく、文字コードによって定義されると解釈できる。文字自身が言語に依存しないため、文字を構成する「記号」と「記号構成ルール」も言語に依存しない。

一般に文字コードは文字を再構成することが可能なので、文字コードのコードポイントは 2.2.1 の「記号」と一致する。これより、文字コードのコードポイントと文字の写像関係は「1 つ以上の複数文字コードポイント : 1 文字」であると定義でき、かつ写像は言語に依存しない。

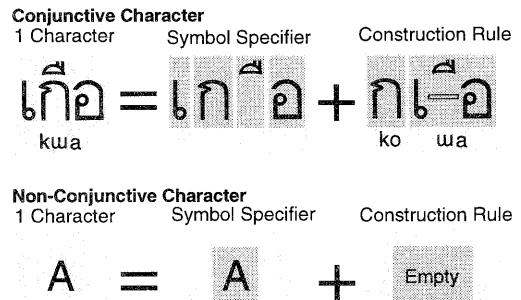


図 2: 構造による文字の分類

2.3 文字と図形・フォントの関係

文字とは概念的な存在であり、使用時には図形により可視化される。一般に文字は特定の 1 図形を示していると考えられているが、例えばアラビア文字では、文字図形は隣接する文字との結合性により決定され、‘(‘) も表記方向 (左から右、右から左) により、開き括弧と閉じ括弧の図形が変化する。すなわち、文字 : 図形 ≠ 1 : 1 である。

文字の図形について分析を行った結果、1 : 1 とならない場合は次の通りである。

表記方向依存図形 開き括弧・閉じ括弧などのように、表記方向によって図形が異なる場合

表記位置依存図形 アラビア文字のように、隣接文字との結合性により図形が異なる場合

リガチャ アラビア文字やデバナガリ文字に見られるように、複数の文字が1図形で表記される場合

変字体 モンゴル文字のように、同一の文字を複数の図形から選択して表記する場合

よって、文字とその図形との関係は「1:1」「複数:1」「1:複数」である。

また、実際にコンピュータ上やプリンタなどに表示を行う場合、フォントを使用する。フォントは、図形をコンピュータまたはプリンタなどの形式に変換したものと定義できる。しかし、フォントはデザイン的な要素も含んでおり、1つの文字図形に対して複数のフォントが存在している。更にデザインによっては、文字図形と1:1に対応しておらず、何らかの写像を行うことも考えられる。したがって、文字図形とフォントもまた必ずしも1:1の関係ではないため、それぞれ別の集合として定義する。

2.4 System 1 の構築

すべてのエンコーディングが混在可能であり、かつすべての文字コードから文字・図形・フォントを統一的に決定することが可能である。また、テキスト処理の対象である文字は言語非依存に定義可能なので、この文字を単位とする処理は言語に依存しない。したがって、POSIX Locale Modelのような地域化環境モデルに依らないため、すべての文字コード・文字を同時に使用可能な国際化環境モデルとして、Global Modelを構築した(図3)。

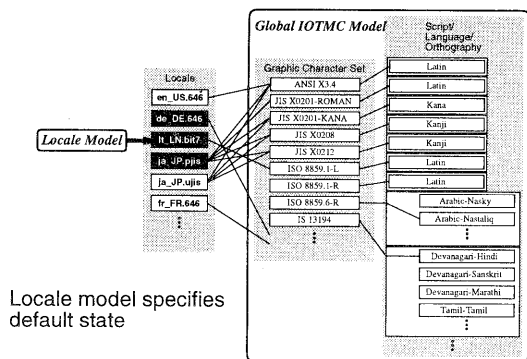


図 3: Global Model

このモデルは、ISO 2022 に準拠し、文字コード・エンコーディングをインタラクティブに切替えることが可能である。また、Locale は Global Model のエンコーディング・指示状態・呼出状態のデフォルトを与えると定義す

れば、既存の Locale への互換性も保持することが可能である。

このモデルに基づき、System 1 をライブラリとして作成した。ライブラリ内部では文字コード・文字・文字図形・フォント間の写像を行う。また、外部プログラムがこのライブラリを利用可能とするために、C 言語関数・Xlib 関数の拡張 API を提供する。

2.5 国際化 Web Browser の設計方針

既存の Web Browser では、同時に使用できるエンコーディングが1種類に限定されていた。しかし、エンコーディングは混在使用が可能であり、かつすべての文字を統一的に処理することが可能である。そこで、System 1 を利用して国際化 Web Browser を作成する。

HTML の仕様では、1文書で使用できるエンコーディングを1つに固定することになっている。すなわち、任意に文字を混在した HTML 文書を作成することができない。これは、XML の利用などにより解決可能と思われるが、現状でほとんどの Web Page が HTML に準拠していることから、Web Browser の対象を HTML 文書とする。そこで、HTML の仕様を拡張し、ISO 2022 に基づいた文字コードの切り替えを可能とする。

なお、HTML 文書でのエンコーディング・言語の指定や、Web Browser の機能としてのエンコーディングの指定を、国際化 Web Browser のエンコーディングデフォルトを与えるると解釈すれば、既存の HTML 文書との互換性は保持可能である。

3 文字列の表示

文字列の表示とは、文字図形を二次元座標空間にフォント配置により表現することである。ここで、ラテンアルファベットとアラビア文字が混在した文字列では、メモリ上のコードポイントの並び順に従った、単純なフォント配置はできない。さらに、縦方向表記や、縦横方向混在をも考慮しなければ、メモリ上で文字のコードポイントを混在使用可能であっても正確に可視化することはできない[9]。

しかし、一般的な出力機構では、左から右へ表記し、上から下へ改行を行うことのみ考慮されており、その他の表記方向や、混在表示については考慮されていない。そこで、任意に文字を混在させて表示するために、フォント配置について考察する。

図形を二次元座標空間上に配置するためには、

- 図形配置開始位置
- 図形配置方向

が必要である。図形配置開始位置・図形配置方向は、ともに、正書法による暗黙上の規則に従っており、アプリケーションまたはユーザにより明示的に与えられる。図形は配置開始位置から規則的に配置されるが、一般に図形配置領域の制限などが更に与えられ、配置開始位置を禁則処理情報などを参照し変更(改行)する必要がある。そこで、改行を含まない1行の表示と、改行を含む複数行の表示の2段階に分けて考察を行う。

3.1 1行の表示

1行の表示とは、指定された配置開始位置から図形をある規則に従って配置することである。複数の文字種を1行で表示する場合、図形の配置方向は左右上下の4方向が存在するので、行内での配置方向混在が生じる。この混在方法として、次が考えられる。

- (1) 物理表記方向を統一した図形列の配置
- (2) 正書法で定められた配置方向を完全に遵守した図形列の配置
- (3) <BDO>、などのタグや、そのタグの属性によって配置方向を指定した図形列の配置
- (4) ISO 6429などの制御機能により配置方向を指定した図形列の配置

ここで、(2)を遵守して表示を行った場合、横書きと縦書きが混在する可能性がある(図4)。また1行が二次元で表現されるとき、図形配置開始位置が変更されるため可読の保証はない。更に、改行方向が不定になるため、複数行表示は不可能である。(4)では強制的に図形配置開始位置が変更可能である。そのため、(2)、(4)による表示はここでは考慮しない。また、(3)により強制的に配置方向を与えることができるが、これは(1)の配置方法に含まれる。そこで、(1)について考察を行う。

例として、ラテンアルファベットとアラビア文字の混在配置を考える。同一配置方向をもつ連続した文字列の単位をブロックと定義すると、ラテンアルファベット列で1ブロック、アラビア文字列で1ブロックを構成する(図5)。このブロックをその行の図形配置開始位置から一次元に配置すれば、配置方向の混在が可能である。正書法によって、文字自身は図形配置方向をもっているので、図形配置開始位置を明示的に与えれば可能である。

すべての配置方向を統一した物理的表記方向で混在する場合、正書法で定められていない物理的表記方向(ラテンアルファベットなら縦表記)で図形列を配置する必要が生じる。このときブロックの配置方向に対し、図形列は固有の改行方向を持つので、複数行の表示を考慮し、デフォルトとしてブロック列の改行方向を一致させるように表示する必要がある(図6)。

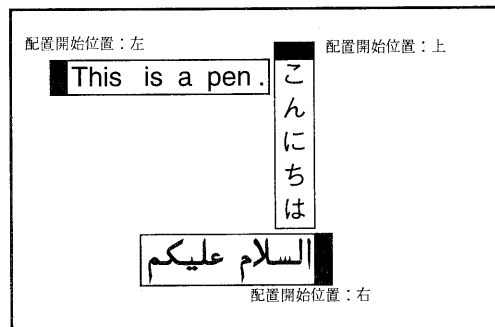


図 4: 1行の二次元配置

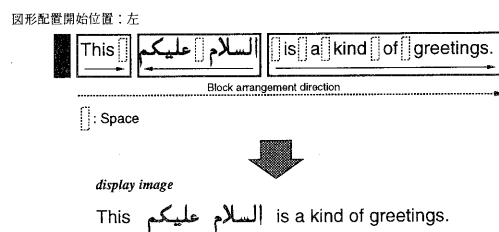


図 5: 1行内の表示方向の混在

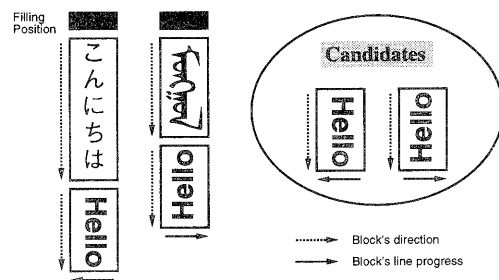


図 6: 横書きと縦書きの混在表示

3.2 複数行の表示

一般に複数行の表示は、左から右へ図形を配置した行の、上から下への改行に固定されており、その表示領域の横幅に合わせて改行する。しかし、1行の表示には横と縦があるため、改行方向も横と縦が存在する。これらの組み合わせは、

$$2(\text{行の縦横表示方向}) \times 2(\text{改行方向}) = 4(\text{通り})$$

である。これは1行の表示においても同様に

$$2(\text{行内表示方向}) \times 2(\text{縦横}) = 4(\text{通り})$$

であり、その表示方法として、行内図形列の同一配置方向を持つ連続した文字列の単位をブロックとして、ブロックの配置を行うことで表示を可能とした。そこで、複数行においても同様に、次の2段階に分けて考える。

- (1) 改行方向が横方向または縦方向に連続している領域を考える。同一改行方向をもつ連続した行の単位を行ブロックとする。このブロックを行配置開始位置から一次元に配置することで、行方向の混在が可能となる。これを複数行ブロックとする。
- (2) 更にブロック配置開始位置を定義し、複数行ブロックをブロック配置開始位置から一次元に配置することで、すべての改行方向が混在可能となる(図7)。

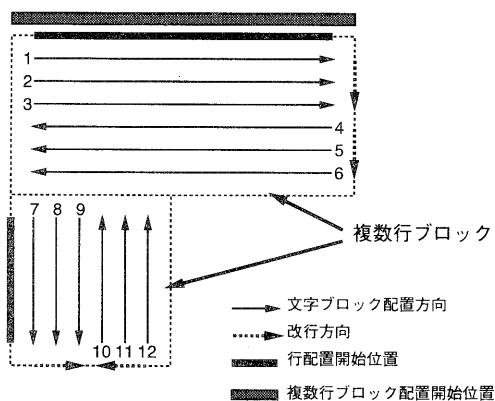


図7: 複数行の配置

3.3 Web Browser の表示機能

任意に文字を混在させて表示するためには、次の情報が必要である。

- (1) 図形配置開始位置
- (2) 改行方向
- (3) 行配置開始位置
- (4) 複数行ブロック配置開始位置

HTML の仕様では、これらを指定する手段が定義されていない。これは HTML の限界であり、XML などの利用を含めた HTML の仕様の再構成が必要である。

そこで、国際化 Web Browser を実現するにあたり、次の機能を実現する。

- <P>、<FRAME>、<BODY> に (1) ~ (4) を指定する属性を追加し、サポートする

- ISO 6429 の制御機能をサポートする
- Web Browser 側のメニューにより (1) ~ (4) を指定可能とする

4 文字列の検索

文字列の検索とは、単に文字の図形で検索を行うことと考えられがちであるが、文字の情報を任意に組み合わせ、検索することも可能でなければならない。例えば、アラビア文字では1文字が複数の図形で表示されるため、単に図形で検索を行うと、異なる図形で表示されている同一の文字を検索することができない。すなわち、検索には、純粋な文字検索と文字図形検索が存在する。文字図形検索では、更に図形名検索とフォント名検索に分類されるが、ここでは図形名検索を対象とする。

検索対象となる文字列は、任意複数の文字種なので、表示図形のほか、表示方向なども考慮する必要がある。そこで、検索のうち最も主要である

- (1) 単一文字種
- (2) 複数文字種の混在

について考察を行う。これらの情報は文字がもつ情報であり、文字コード (Multibyte Character; mb) から変換された文字 (Wide Character; WC) を使用し、処理を行う。WC は、文字の ID のほか、文字のもつ図形情報 (表記方向依存図形情報など)、表記情報 (図形配置開始位置など) を含んだ固定長のコードである。

4.1 単一文字種による文字列の検索

単一文字種では、文字の表記方向が同一なので、考慮する必要がない。そこで、文字検索と図形検索の方法について考慮すればよい。

文字検索 純粋に文字のみの検索なので、対象となるのはその文字の ID (WC-ID) だけである。

図形検索 図形を考慮するので、WC-ID の他、文字が持っている図形情報として、表記方向依存図形情報、表記位置依存図形情報、リガチャ情報、変字体情報も参照して、検索を行う。

4.2 複数文字種の混在による文字列の検索

複数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可能性があり、その場合の検索文字列は、単純にメモリイメージと対応しない場合がある。図形配置開始位置によりさらに表示イメージが異なる。表示イメージに合わせた検索を行った場合、検索結果はメモリイメー

ジとは一致しない (図 8)。文字図形は図形配置開始位置・文字表記方向に従って表示されるので、表示イメージに合わせた検索では、単純な図形名検索だけではなく、図形配置情報を考慮する必要がある。

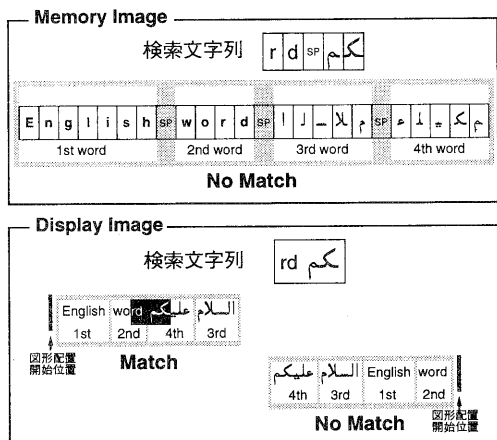


図 8: 検索文字列と被検索文字列のずれ

検索文字列は文字列の表示上の連続性を考慮すれば図形配置開始位置を 1 つだけでもつので、被検索文字列の図形配置開始位置を考慮すればよい。そこで、次に文字表記方向を考慮した検索について考察する。

同一の表記方向をもつ検索文字列 検索文字列の始点となる文字から終点となる文字まで連続的に並んでいるため、前述の単一文字種による文字列の検索と同様に検索可能である。

表記方向が混在した検索文字列 検索文字列の始点となる文字から終点となる文字まで連続的に並んでいない。図 9 では、各文字の表記方向は矢印で示した通りで、終点に見える文字は、アラビア文字部分の先頭の文字であり、検索文字列の中間に存在している。これを検索する場合、ラテンアルファベットとアラビア文字の間に他の文字が存在する可能性がある。そこで、検索文字列を、同一表記方向をもつ文字列ごとに検索ブロックとして区切っておく。このとき、検索ブロック間の境界で文字の表記方向が変化するので、先頭の検索ブロックを検索し、マッチした段階で次の文字の表記方向を検査する。表記方向が異なる場合、次の検索ブロックを検索する。検索中被検索文字列の表記方向が変化した場合は、その文字列が含まれていないことになる。すべてのブロックがマッチした場合、検索文字列がマッチしたことになる。

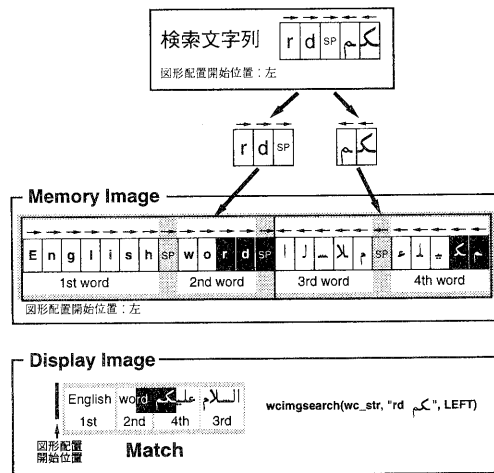


図 9: 表記方向が混在した文字列の検索

4.3 HTML のタグを考慮した検索

HTML 4.0 では、ラテンアルファベットとアラビア文字など、横書きで表示方向が異なる文字種の混在表示を実現するため、ブロックエレメントとして <BDO> タグと タグが用意されている。このタグを使用すれば、表示の際にフォントの配置方向を変更することができる。このタグを使用している文字列を検索する場合、HTML パーザによって、文字列は分割されるが、実際は連続した文字列であるので、基本的に前述の表記方向が混在した検索文字列と同様に検索可能である。

4.4 Web Browser の検索機能

文字列検索では文字検索と文字図形検索を考慮する必要がある。また、表記方向が混在した文字列も存在するので、図形配置情報も考慮する必要がある。そこで国際化 Web Browser では次の検索条件を与えることを可能とする。

- 文字検索と文字図形検索の指定
 - 表示方向依存図形情報の指定
 - 表示位置依存図形情報の指定
 - リガチャ情報の指定
 - 異体字情報の指定
- 図形配置開始位置の指定

