CharsetWiki: バザールモデルによる文字集合定義システムの実装

江渡 浩一郎

要旨

符号化文字集合に依存しない文字処理モデルとして CHISE Project において Chaon モデルが提唱され、XEmacs CHISE、Ruby/CHISE、Perl/CHISE などの処理系によって実用化が進められている。しかしこれらは主に開発者向けのライブラリや、技術力が高いユーザ向けのシステムである。そこで、一般の利用者がこれらの処理系を容易に使えるようにするために、Web 上の共同編集システム (WikiWikiWeb) を元に共同で文字定義を編集できるシステム “CharsetWiki” の実装を行なった。このシステムにより、独自に必要な文字集合を定義し、使用することができる。

キーワード：CHISE Project, Chaon model, Open Character Set, Bazar model, Entity Reference, WikiWikiWeb, Ruby/CHISE, KAGE

1 背景

1.1 符号化文字集合の限界

符号化文字集合によって文字を処理する方法について、過去に様々な議論がなされてきた。特に符号化の対象から外れた文字をどのように扱うかについて、様々なアプローチがある。

1. 外字の利用
2. 大規模文字集合
3. CHISE Project

このそれぞれについて、順に説明を行う。

1.2 外字によるアプローチ

通常の文字処理方法においては JIS (Japan Industrial Standard) や ISO (International Organization for Standardization) といった公的な組織が策定した文字集合規格を用いる。そのような符号化の対象から外れた文字は、外字と呼ばれる仕組みによって処理することになる。文字コードのある領域をあらかじめ外字領域として割り当てておき、そこにそのシステムで使いたい文字を個別に登録していく。この外字によるアプローチには様々な問題がある。

- それぞれのシステムで独自に定義するため、他のシステムと交換可能でない。
- 文字の同定に必要な情報が付与されていない。
- 文字が画像として扱われている。

それぞれのコミュニティが独自に文字集合を定義する例もある。例えば各地方自治体では戸籍情報の管理のために、既存の文字集合に含まれない人名に要する漢字などの文字に独自の符号を与えている。また、印刷会社も独自に文字集合を管理している例がある。このような独自の符号は他のコミュニティと交換することができないため、大きな問題となる。

1.3 大規模文字集合

このような文字不足に対処する一つの方法は、単純に文字数を増やすということである。大規模な文字集合の例として、文字鏡、TRON コード、Unicode、JIS X 0213 などがあげられる。

文字鏡 [2] ではその基本となる文字集合として諸橋大漢和辞典 [3] に含まれる漢字を全て扱っている。この字書には 5 万字以上の親字[1]が収められ、それぞれに対して大漢和検字番号という番号を振っている。そのためこの番号を元に符号化を行えば、大規模な符号化文字集合の作成を行えるのではないかという発想が生まれた。文字鏡ではさらに甲骨文字、西夏文字、変体仮名などの文字を追加している。また未収録文字の作字・登録を一般に募集している[2]。

TRON コード [5] は、様々な文字集合の和集合として成り立っており、諸橋大漢和辞典の文字集合も含んでいる。社団法人 TRON 協会内の TRON 文字収録センターが文字登録の募集を行い、検査の後に追加を行っている[3]。

Unicode[6] では日中韓の同じ形を持つ漢字を統合 (Han Unification) するという形で漢字の文字数が減ら

[1] 字書の見出しに相当する字

[2] そのようにして追加された文字は、文字の出典などの典拠が一般には公開されていないという批判もある [4]。

[3] 実世界で使われている文字だけではなく創作による文字も収録を行う方針をとっており、実際に森岡浩之氏による小説「星界シリーズ」にて使用されている「アーブ文字」を文字集合として登録している。

されてはいるが、それでも BMP (Basic Multilingual Plane) だけで漢字は約 21,000 文字以上あり、またその後 Ext.A、Ext.B という形で多数の漢字が追加されている。

JIS においても、JIS X 0213[7] によって現在の日本で実際に用いられている漢字の網羅的な追加登録が行なわれ、これにより日本の現実的な状況で必要とされる文字はほとんど収録されることとなった。

このように大規模文字集合の出現と実用化によって、文字集合に登録されていないために外字としてしか使えないという状況は大幅に軽減された。しかしどれだけ大量の文字が追加されたとしても、本当に使いたい文字がそこに含まれているという保証はない。また始めからどの文字集合にも属していない文字や、自分独自の文字を使いたい場合などは外字という手法に頼らざるをえない。

必要に応じて自分独自の文字を作り、それが後に一般化することによって漢字という文化が発展してきたという面もある。例えば「鞄」という漢字はカバンを表す文字として現在では一般的だが、元々は銀座タニザワの創始者谷澤禎三氏が明治十二年に開店したカバン店の看板に、当初は「革包」と表記していたものを後に合字によって「鞄」に変えたところ、その合字が普及したものであると言われている [8]。

文字を符号化して扱う方法では自分独自の文字を作ることは非常に困難であり、結果としてこのような発展の可能性をあらかじめ排除してしまっている状況となっている。自分独自の文字を他の文字と共に扱い、他人と共有できるような環境を提供することが、漢字文化の発展に寄与できると考えている。

1.4 CHISE Project

CHISE Project は、既存の符号化文字集合に依存した文字処理方法からの根本的な脱却を目指すプロジェクトである。プロジェクトの創始者である守岡知彦氏によって符号化文字集合に依存しない文字処理モデルとして Chaon モデルが提唱された [9][10][11]。

Chaon モデルとは「文字を素性 (そせい)[4]の集合によって定義する方法」である[5]。例えば既存の符号化文字集合においては、「字」という文字は JIS X 0208 文字集合における 1 面 27 区 90 点にある文字として符号によって指定する。しかしそれ以外にもこの文字を指定する方法にはいろいろあるはずである。例えば「ウカンムリの下に子がつく文字」のように形で指定する方法や、部首、画数、読みなどの複数の素性の組み合わせで指定する方法もありうる。このように様々な素性の集合によって文字を指定することができるという方法が Chaon モデルである。

この Chaon モデルという概念に基づき、XEmacs CHISE (編集環境)、Ruby/CHISE、Perl/CHISE (プログラミング・ライブラリ)、Ω /CHISE (多言語組版環境) などのソフトウェアが実装され、全てフリーソフトウェアとして公開されている。

このような仕組みによって、従来は文字コードに依存した文字処理しか行えなかったものを、CHISE が提供する環境においては、素性に基づく文字処理を行えるようになった。

しかしこれらは、主に開発者や高度な技術者が使うことを想定したシステム[6]であり、現在はまだ一般の利用者が使うのは困難である。

2 CharsetWiki

このように Chaon モデルを元にした CHISE Project におけるソフトウェアは、高度で柔軟な文字処理環境を提供しているが、気軽に用いることができるものにはまだなっていない。この問題を解決するために、だれもが Web 上で簡単に文字定義を行い、その文字をその場で使用することができるシステム “CharsetWiki” を開発した。

この CharsetWiki は「バザールモデル」に基づき、多数の人が文字集合定義に参加できるものである。

2.1 バザールモデル

バザールモデルとは、エリック・レイモンドが「伽藍とバザール」[12] において提唱したソフトウェア開発スタイルの一種である。ある特定の比較的少人数の人々がソフトウェアの仕様を策定し、開発するというモデルを伽藍型モデルと言う。それに対して、多数の開発者が同時に開発に参加し、そこで各々の役割を果すというモデルをバザールモデルという[7]。

今回の文字集合定義の問題に置き換えると、JIS や ISO といった公的機関が策定した文字集合を伽藍型

[4] 素性とは属性に非常に近い概念であるが、すでに定義された文字が保有する情報を「属性」といい、まだ定義がなされていない文字を定義するための情報が「素性」である。つまり素性を束ねたものが、文字の定義となる。

[5] Chaon モデルの前身として UTF-2000 モデルがある。これは Chaon モデルとほぼ同じものを意味しているが、その名称から文字コードのエンコーディング方式を意味しているという誤解を受けることがあり、名称を変更した。

[6] XEmacs CHISE は Unix 上でのシステムであり、また大量のフォントのインストールが必要である。Ω は LaTeX を基盤としたソフトであり、LaTeX を使いこなせることが前提である。

[7] 「伽藍とバザール」では、前者の例として GNU 型の開発モデル、後者の例として Linux 型の開発モデルをあげている。GNU ソフトウェアも Linux も、どちらも GPL (GNU General Public License) をライセンスとして採用したフリーソフトウェアであり、このようにライセンスの問題は切り離して論じられているが、現実の状況においてはライセンス形態も大きな問題となる。

モデルと考えることができる。文字鏡や TRON コードにおいても公的な規格ほど厳格ではないが文字を登録するセンターが存在しており、小規模な伽藍型モデルであると考えられる。

それに対して特に中心というものが定義されずに、各自が文字集合の定義に参加できるモデルを、文字集合のバザールモデルと考えることができる。Wiki のようなだれでも自由に書き換え可能な基盤を用いれば、このような文字集合のバザールモデルが可能となる。

また、CHISE Project において公開されているソフトウェアは全てフリーソフトウェアであり、自由に改変・再配布可能である。このような基盤においてこそ文字集合のバザールモデルは可能になると考えている。

このようにバザールモデルを使用することによって、だれでも自由に文字集合の定義を行えるようになる。

3 実装

CharsetWiki の実装を要素毎に示す。

1. 実体参照の拡張 ; 独自に作成する文字を指定する方法として、XML における実体参照という枠組を拡張した形式を使用する。
2. WikiWikiWeb ; Web 上の共同編集の基盤として WikiWikiWeb を使用し、その機能拡張として “Charset Plugin” を実装し、それにより文字定義と文字参照を行う。
3. Ruby/CHISE ; 個々の文字に対して必要に応じて素性を定義し、文字データベースに保存することができ、この機能が “Charset Plugin” の基盤となる。
4. KAGE ; IDS によって指定された文字の自動合成を行い、それを画像として生成するために利用する。

以下、それぞれの実装について詳しく述べる。

3.1 実体参照の拡張

一般的な文字処理環境は、ある特定の符号化文字集合から外れた文字は使用できない。XML において、そのような文字を指定するために実体参照[8]という仕組みが用意されており、Unicode のコードポイントを直接指定することによってその文字を指定することができる。

文字鏡と TRON コードでは、この実体参照という枠組を独自に拡張して文字の指定に用いている。例えば文字鏡では「`&Mnnnnnn;`」という形式で、文字鏡番号を直接指定することができる。また TRON コードではこれをテキスト形式 TRON コードという名称で呼んでおり、「`&T-xxyyyy;`」という形式で TRON コードを直接指定できる。

CHISE の実装においては、また別種の実体参照の拡張を行なっている。例えば、「`&J90-xxxx;`」という参照は、JIS X 0208:1990 の文字集合におけるコードポイントを直接指定したものと解釈する。同様に、「`&J83-xxxx;`」や「`&J78-xxxx;`」といった実体参照は、それぞれいわゆる 83JIS、78JIS のコードポイント[9]を指す。またそれ以外にも、CDP や CBETA などの様々な文字コードを実体参照として用いることができる。しかし XEmacs CHISE は、通常のエディタとは異なり、実体参照で表わされた文字も他の文字と同様に表示・編集の対象とすることができる。

今回はこれらの実体参照の仕組みをさらに拡張し、自分独自の文字集合の参照に用いている。

3.2 WikiWikiWeb

WikiWikiWeb(略称 Wiki)[13] は、書き換え可能な Web システムである。Web で見ているそのページを、その場で Web ブラウザ経由で書き換えることができる。簡単な記法によって構造を持った文章を書くことができ、また新しいページを作り、ページからページへのリンクを張ることができる。

このテキストは、HTML を簡単にしたマークアップ記法 (Wiki 記法) によってヘッダー、リスト、テーブルなどの構造を示す要素を記述することができる。図 1 に Wiki 記法の一例を示す。この例にあげた記法においては、行の先頭に「*」がある行はヘッダー、「-」はリスト、「:」は用語説明を表す。行頭に「|」がある行はテーブルを表し「|」が項目間の区切りとなる。このように、行頭の一文字を見るだけでどのようなマークアップかを判別することができ、行のパースが容易になっている。図 1 に示すようなブラウザ上

[8]実際には実体参照という枠組の中の文字参照と呼ばれる仕組みであるが、一般には特に区別されずに用いられることが多い。

[9]JIS X 0208 とはいわゆる JIS 漢字を規定した規格である。83JIS、78JIS(新 JIS、旧 JIS ともいう) とは正確には JIS X 0208:1983、JIS X 0208:1978 と表記し、つまりその規格が制定された年を表す。通常の規格であれば規格の改変は互換性があるように行われるため年号を付けて区別する必要はない。しかし 83JIS は、78JIS における文字のコードポイントを入れ替えたり、文字を追加するなどといった非互換の改変を行なったため、利用者は現在使用している JIS 漢字が 1983 年のもの (新 JIS) か、1978 年のもの (旧 JIS) かを常に区別する必要が生まれた。これがいわゆる新旧 JIS コード問題である。

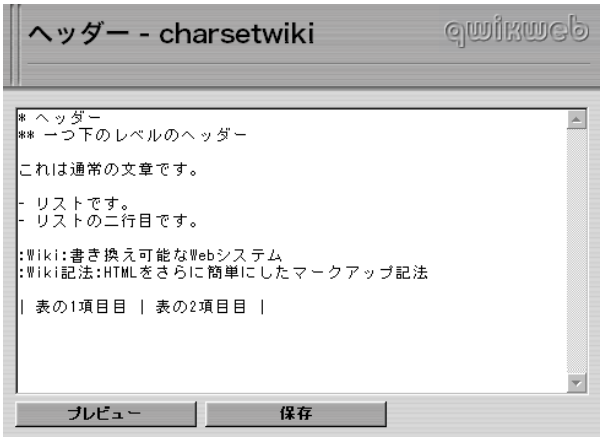

図 1: Wiki の編集画面

図 2: Wiki の表示画面

の編集画面にて編集を行い、そこで保存ボタンを押すと、図 2 のような Web ページとして表示される。

Wiki では、見ているページはだれでも自由に書き換えられる[10]。Wiki そのものは柔軟な仕組みであり様々な用途に使えるが、特に複数人で一つの情報を編集する用途に向いている。例えばオフィスやグループ内の情報共有に用いられることが多い。一方、百科事典の編集というもっと大きな対象に Wiki を利用している WikiPedia[15] という例もある。これはバザールモデルによって共同で百科事典を作りあげようという壮大な目標を持った Wiki サイトである。このような事例から、文字集合のような情報を Wiki を用いて編集することもできるのではないかという発想が生まれた。

現在、元々の Wiki から派生した様々な特徴を持つ[11]多数の Wiki エンジンが開発され、その多くはフリーソフトウェアとして入手可能である。特にプラグインという仕組みによって機能を拡張可能としているものもある。これにより、Wiki エンジンに動的に機能を追加していくことができるようになった。

今回は Hiki[16] という Wiki エンジンを元に筆者が独自に拡張した Wiki エンジンを基盤として用い、そこに Charset Plugin という文字定義を行うプラグインを実装することにより、Wiki 上で文字集合定義を行う CharsetWiki を実現した。

3.3 Ruby/CHISE

Ruby[17] とは、まつもとゆきひろ氏によって作られたオブジェクト指向スクリプト言語である。言語の根本的な部分からオブジェクト指向で設計されており、また同時に Perl のように簡単に書ける記述性の高さを合わせ持つ。フリーソフトウェアとして配付されており、日本のみならず世界的にも人気が出ている[12]。

Ruby/CHISE[18][19] とは過去に筆者が制作した Ruby における Chaon モデルの実装である。Ruby では、例えば String クラスのような言語の根本的な部分を、特異クラスと呼ばれる動的な仕組みによって変更できる。Ruby/CHISE はその仕組みを利用し、あたかも文字列そのものがオブジェクトであるかのように機能し、文字そのものに対してメソッドを発行することができる。図 3 に、「字」という文字の各文字集合における文字コードなどを表示する例を示す。

内部の実装としては、個々の文字は背後にある文字データベースへと接続されており、その文字自身をキーとしてその情報にアクセスし、その結果となる情報を返している。

さらに、それぞれの文字に対して素性の定義を行うこともできる。例えば図 4 にあげるプログラムを実行すると、「木」という文字の mydepth 素性を 1、「林」を 2、「森」を 3 として定義している。ここで定義された素性は自動的に文字データベースに保存され、その後にプログラムを実行した際にその値を取り出すことができる。この独自の素性を必要に応じて定義し、文字データベースに保存することができるという機能が、CharsetWiki の基盤となっている。

[10] これを非常に危険だと感じる人も多い。しかしむしろ自由に書き換えができるから荒らす気も起きなくなるという面もあり、実際にいままで多数の Wiki サイトが、だれでも自由に書き換えられる状態を継続し、発展し続けている [14]。

[11] Wiki 記法で用いられる記号と構造の対応関係は個々の Wiki エンジンにおいて異なっており、ユーザの混乱を招いている。これは Wiki の欠点の一つである。

[12] 例えば、Apple 社の Mac OS X(v.10.2 以降) においては OS の一部として標準搭載されている。

```
require 'chise'
include CHISE
p "字"                    # "字"
p "字".ucs                # 23383
p "字".daikanwa           # 6942(大漢和番
号)
p "字".total_strokes      # 6
```

図 3: Ruby/CHISE の基本的な使い方

```
"木".mydepth = 1
"林".mydepth = 2
"森".mydepth = 3
```

図 4: Ruby/CHISE における素性定義の例

3.4 KAGE

KAGE[20] とは、上地宏一氏による文字グリフ自動合成システムである。文字合成を指示する IDS (Ideographic Description Sequence) を URL 中に独自のエンコードで埋め込み、インターネット上の KAGE サーバにリクエストを送ると、その IDS に応じた文字グリフを自動的に合成し、結果として PNG や SVG などの画像が取得できる。文字デザインの種類として、ゴシック、明朝を選ぶことができる。

IDS とは、文字合成を指定するための文字列を指定するための Unicode において規定された標準規格である。漢字は主に偏と傍などの部品によって構成されるが、そのような合成の種類を IDC (Ideographic Description Characters) と呼ばれる文字合成の種類を規定するオペレータで指示し、その後に合成したい文字を順に並べると、それが IDS となる。IDC は、Unicode のコードポイント U+2FF0 ~ U+2FFB で示される文字であるが、その文字を簡単に入力する手段がないため、本システムでは代替記号として#0 ~ #B を使えるようにしている。

本システムでは IDS によって指示された文字の形態を画像とするにあたって、KAGE を文字画像化エンジンとして利用している。

3.5 CharsetWiki の利用例

実際に Wiki 内で独自の文字集合を定義し、使用する例をあげる。例えばある一つのページがあり、そこに二つの独自に定義した文字が埋め込まれていると

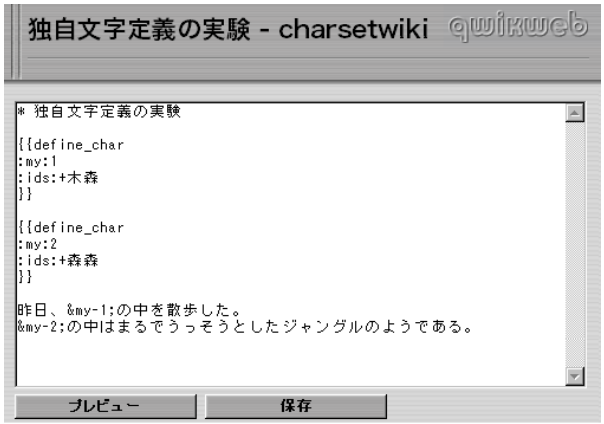

図 5: CharsetWiki の編集画面

する。

図 5 が、Wiki 記法によるテキストである。まずテキスト前半の「{{define_char ... }}」に囲まれたエリアが Charset Plugin への指令を表す。その中に記述された「:my:1」という行で、ここでこれから定義しようとしている文字が my 文字集合の 1 番であるということを指示する。その次の行「:ids:#0 木森」において、その文字の IDS を指示する。このようにして、my 文字集合の 1 番が「#0 木森」という IDS を持つということを定義することができた。同様に、次の指定で my 文字集合の 2 番が「#0 森森」という IDS を持つということを定義している[13]。

次に、その下の部分で「&my-1;」という記述がある。これが my 文字集合の 1 番の文字を表示することを指定している。同様にその次の「&my-2;」において my 文字集合の 2 番の文字の表示を指定している。

実際の Web ブラウザ上の Wiki サイトの画面例をあげる。図 5 が Wiki の編集画面であり、中央のテキストエリアでテキストを編集し、「保存」ボタンを押すとページの内容が書きかえられ、図 6 のような表示画面となる。

3.6 CharsetWiki の内部構造

まず Wiki エンジンはテキストをパースし、define_char プラグインを呼び出す。プラグイン中のテキストを読み、素性名と内容とを認識する。「my」という素性名は特別処理をし、ある特定の文字集合を指すものと解釈する。内部では、my 文字集合は

[13] 実際にはここで独自に定義した&my-1; という文字は Unicode Ext.B にすでに定義されており、また&my-2; という文字もほぼ同じ文字が定義されている。そのため Ext.B を使えるシステムであれば独自に定義する必要はないが、サンプルとしてあげているものである。

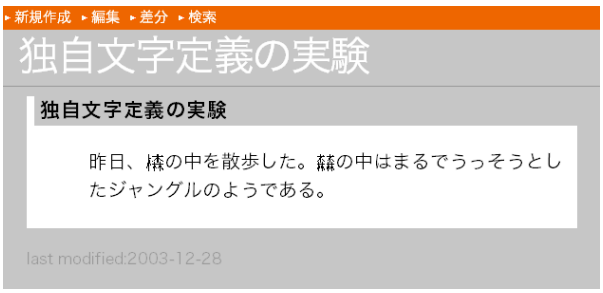

図 6: CharsetWiki の表示画面

Unicode の私用領域 (U+E000 ~ U+F8FF) に割り当てている。つまり内部的には my 文字集合の 1 番は、U+E001 と同じ実体を指す。ただしこの内部コードの割り当ては任意に変更可能である。この U+E001 という文字をキーとし、文字データベースの「ids」という素性に「#0 木森」という IDS を格納する。文字データベースには Berkeley DB[21] を使用し、一素性が一ファイルに対応する形で格納されている。

次に「&my-1;」という記述が文字の表示を指示する。内部的にはまず「{{my_char(1)}}」というプラグインへと置換し、その次にパーザが解析する。つまり実体参照形式はプラグインの省略記法として定義している。そして文字データベースに接続し、U+E001 の IDS 素性を取得する。

次にその IDS を KAGE が認識する URL フォーマット[14]にエンコードする。例えば「#0 木森」という IDS であれば、

http://KAGE サーバの URL/u2ff0u6728u68ee.png

という URL となる。これを画像として埋め込むために HTML を出力する。このとき style 指定で画像の幅を一文字分と指定している。このようにすると、通常の文字と同じ横幅で出力され、ブラウザの機能で文字を拡大・縮小しても大きさは追従するようになる。

<img src="http://KAGE サーバの URL/
u2ff0u6728u68ee.png" style="width:1em;"/>

4 関連研究

4.1 外字交換方式

外字を複数のシステム間で交換可能なものにするという試みもある。EGIX (Embedding Glyph Identifiers in XML Documents)[22](JIS TR X 0047[23] および XKP GAIJI 交換仕様書 [24] がこの規格の元となった) においては個々の外字について様々な情報をサーバ側に保持し、XML 形式で交換することによって外字に関する情報の交換を可能としている。

しかしこの規格に対応したシステムは、地方自治体のような大規模な組織における利用を対象として考えられたものが多く、一般の人が気軽に用いることができるようなシステムは存在していない。

4.2 CDL (Character Description Language)

Tom Bishop 氏と Richard Cook 氏による CDL (Character Description Language)[25][26][27][28] は部品を配置することによって文字を指定するための XML 言語である。漢字の合成に必要な要素をストロークにまで分解し、その合成として文字を組み立てていくことができるため、IDS のような部品を並べる合成方法よりも精度の高い合成を指示することができる。

Tom Bishop 氏による外国人のための漢字学習ソフト「Wenlin 3.1」[29] ではこの CDL が実装されており、文章中に通常の文字と同じように CDL による合成文字を挿入することができる。これにより独自の文字をその場で定義しつつ使用することができる。

独自の漢字をその場で定義し用いることができる点で、Wenlin は本システムに非常に近いが、CharsetWiki では作成した文字をネットワーク上で共有し、共同で編集を行うシステムを含めて考えている点が異なる。CharsetWiki では文字の合成を指示する方法として IDS を用いているが、それと共に CDL を使えるようにすることは非常に有効であると思われる。

5 議論

ここでは一つの Wiki サイト内に閉じた文字定義のシステムを提示した。このようにして定義した文字集合を他のサイトから利用する手段は現時点では用意していないが、Wiki には InterWikiName という Wiki サイト間の接続を指示するための仕組みがあり、この仕組みを使用して他の Wiki サイトの文字定義情報を使用することが考えられる。

現在の XEmacs CHISE、Ruby/CHISE などは、その文字データベースをそれぞれローカルのマシン上に保持している。しかしこの文字データベースはネットワーク経由でアクセスでき、またネットワーク経由で更新可能だと理想的である。このようなデータベースシステムの実装の基盤として CharsetWiki を使用する可能性があると考えている。

[14]KAGE の URL フォーマットは現在 v.0.3 から v.0.4 への移行期にあるが、この例では v.0.3 のものを使用している。

6 結論

Chaon モデルに基づいた符号化文字集合に依存しない文字処理環境として、WikiWikiWeb を基盤とした Web ブラウザから容易に文字集合定義を行い、使用することができるシステム “CharsetWiki” を実装した。従来の Chaon モデルに基づいたシステムは、開発者や技術力の高いユーザー向けだったが、これを一般の利用者が Web ブラウザ経由で容易に利用可能なものとした。これにより、独自に必要な文字を定義し、その場で利用できることを確認した。

謝辞

守岡知彦氏、上地宏一氏、g 新部裕氏、藤原義久氏、鈴木泰博氏、塚田浩二氏に感謝の意を表します。

参考文献

[1] 芝野耕司編著『JIS 漢字字典』日本規格協会、2002.

[2] 文字鏡研究会編『パソコン悠悠漢字術〈2002〉今昔文字鏡徹底活用』紀伊国屋書店、2002. http://www.mojikyo.org/

[3] 諸橋轍次編『大漢和辞典』大修館書店.

[4] 山田崇仁「ソフトウェアレビュー 今昔文字鏡」『漢字文献情報処理研究』第 3 号、pp.150、漢字文献情報処理研究会、好文出版、2002.

[5] TRON 文字収録センター、http://www2.tron.org/

[6] “Unicode 4.0 The Unicode Standard”, Addison-Wesley, 2003.

[7] JIS X 0213:2000『7 ビット及び 8 ビットの 2 バイト情報交換用符号化拡張漢字集合』日本規格協会、2000.

[8] 「誰が文字を作るのか?」『JIS 漢字字典』p.229、日本規格協会、2002.

[9] CHISE Project、http://cvs.m17n.org/chise/

[10] 師茂樹「CHISE プロジェクト」『漢字文献情報処理研究』第 3 号、pp.143-146、漢字文献情報処理研究会、好文出版、2002.

[11] 守岡知彦、江渡浩一郎、苫米地等流、宮崎泉、師茂樹「CHISE Project」『漢字文献情報処理研究』第 4 号、pp.58-69、漢字文献情報処理研究会、好文出版、2003.

[12] エリック・S・レイモンド、山形浩生訳『伽藍とバザール』光芒社、1999.

[13] Bo Leuf、Ward Cunningham 著、yomoyomo 訳『Wiki Way—コラボレーションツール Wiki』ソフトバンクパブリッシング、2002.

[14] Ward Cunningham による最初の Wiki サイト、http://c2.com/cgi/wiki

[15] Wikipedia、http://en.wikipedia.org/ Wikipedia 日本語版、http://ja.wikipedia.org/

[16] Hiki、http://www.namaraii.com/hiki/

[17] Ruby、http://www.ruby-lang.org/

[18] Ruby/CHISE、http://eto.com/2003/ruby/

[19] 守岡知彦、江渡浩一郎「文字知識処理環境 CHISE の基盤ソフトウェア」Linux Conference 2003、2003.

[20] 上地宏一「漢字フォント自動生成サーバ〝影 KAGE〟の構築 – 文字コードの枠組みを超える次世代漢字処理の提案– 」、『漢字文献情報処理研究』第 3 号、pp.4-13、漢字文献情報処理研究会、好文出版、2002.

[21] Berkeley DB、http://www.sleepycat.com/

[22] Embedding Glyph Identifiers in XML Documents, W3C Note, 2002. http://www.w3.org/TR/EGIX

[23] XML による画像参照交換方式, TR X 0047:2001. http://www.y-adagio.com/public/standards/tr_lsi_xml/l

[24] 拡張漢字処理協議会、XKP GAIJI 交換仕様書 第 1.0 版、2000. http://www.xkp.or.jp/next/XKPGAIJI100.htm

[25] CDL (Character Description Language)、http://www.wenlin.com/cdl/

[26] Tom Bishop, Richard Cook, “Character Description Lang. Spec.”, IRGN985 CDL Specs, ISO/IEC JTC1/SC2/WG2/IRG, 2003.

[27] Richard Cook, “Chinese Character Description Lang”, IRGN986 ChineseCDL, ISO/IEC JTC1/SC2/WG2/IRG, 2003.

[28] Tom Bishop, Richard Cook, “Character Description Lang. (Stroke Types)”, IRGN987 CDL Strokes, ISO/IEC JTC1/SC2/WG2/IRG, 2003.

[29] Wenlin Institute、http://www.wenlin.com/

CharsetWiki: Implementation of Character Set Definition System based on Bazaar Model

Kouichirou Eto

Abstract

We, CHISE Project, proposed character processing model, called “Chaon model”, that does not depend on Coded Character Set Model. And we developed several system software (XEmacs CHISE, Ruby/CHISE, Perl/CHISE, etc.) to make realize the Chaon model. But, the systems are mainly for developers and experts. Thus, I developed a system called “CharsetWiki” based on collaborative editing system (WikiWikiWeb), that will enables end users to use the CHISE system. You can define a character you wish, and use the character.