3L-4

Ruby における多言語処理環境に関する検討

島田 拓也　伊藤 一成　Martin J. DÜRST
青山学院大学 理工学部

1　はじめに

近年，Web 上でのファイルや文字データのやり取りが盛んである一方で文字コードが数多く存在する．国際化の観点から考えると，各プログラミング言語やそれによって実装されたアプリケーションには，各文字コードに対応できる環境が要求される．

ユーザは文字コード変換処理を意識したプログラミングを行う必要があるが，もしプログラミング言語がそのために複雑な記述を要求するようであれば，ユーザの負担は増すばかりである．プログラミング言語としては文字コード変換処理を簡潔に記述可能な環境が整っていることが望ましい．

各プログラミング言語によって，採られている方策と国際化における完成度が異なる．その中で多言語化への対応がまだ初期の段階にある Ruby に注目した．

Ruby は日本で生まれた言語であるため，他言語に比べ日本語文字コードへの対応は重視されている．しかし，それらと日本語以外の文字コード間の連携は不十分である．また，簡単にエンコード処理を記述出来る環境を整えることも必要である．これらを整備することは Ruby の国際化における重要な課題であり，Ruby の更なる普及と発展につながるはずである．

本稿では，Ruby における文字コード変換処理環境の向上を目指すため，現在の文字コード変換ライブラリに対する考察を行い，それらを画一的に利用できるライブラリを提案する．

2　Ruby の現状

近年の Ruby の人気は高く，近いうちに Java のようなメジャーな言語に取って代わるくらいまでになると言う人もいる [1]．世界的に大事な役割を果たす言語になるためには，国際化への取組みを本格化することが大変重要である．

2.1　Ruby における多言語処理

様々な文字エンコーディングに対する多言語化の対象としては，相互変換，テキスト処理，入出力処理などが考えられる．

Ruby のテキスト処理は，Java や Perl，Python などとは方式が異なる．後者では，渡された文字データをより大きな文字レパートリを持つ Unicode に変換してから処理をし，指定された文字エンコーディングで返す方式を採っている．

一方 Ruby では，String クラスのテキストをバイト列として扱い，文字単位の処理は正規表現を用いて行う．また，特定の文字エンコーディングの物理表現を参照する基本的な処理（プリミティブ）を定義し，文字列処理をそのプリミティブを通じて行う方式にすることが今後の方針である [2]．しかし，世界の文字に対応するために機能を増やすとき，この方式でどこまで通用するのか疑問が残る．

本稿では，よりユーザに近い部分である文字エンコーディングの相互変換に注目した．文字エンコーディング変換処理をユーザが記述しなければならない限り，ユーザビリティを向上させることは重要である．

2.2　文字エンコーディングの相互変換

Ruby においてテキストの文字エンコーディングを変換する場合には，ライブラリを用いる．Ruby 1.8 で標準添付されている文字エンコーディング変換ライブラリには Iconv，Kconv，NKF があり，それ以外には Uconv が使われている [3]．

ほぼ同等の機能を有するライブラリが複数存在することは分かりづらい．実際にはそれぞれ微妙な機能の違いがあるので，状況によって使い分けることになる．しかし，本来それはプログラミング言語が担う役割であって，ユーザに負担させるものではないと考える．そこで，現在標準で添付されている文字コード変換ライブラリ群に対し，それらを画一的に利用できるライブラリを新しく実装した．

3　ライブラリの設計と実装

3.1　日本語文字コードへの対応

国際化として日本語以外の文字コードをサポートする環境は，すでに GNU の iconv ライブラリ [4] が Ruby から標準で利用できることで実現されている．しかし，iconv は Shift_JIS と EUC-JP や ISO-2022-JP の相互変換において，ユーザの期待とは異なる動作をする場合がある [5] などの問題を抱えているため，日本語文字コードの変換は NKF ライブラリ（バージョン 2.0）を利用している Kconv ライブラリに任せる．図 1 はその処理のフローチャートである．

A Study Concerning Multilingual Processing in Ruby
Takuya SHIMADA, Kazunari ITO and Martin J. DÜRST
Department of Integrated Information Technology, College of Science and Engineering, Aoyama Gakuin University
5–10–1 Fuchinobe, Sagamihara, Kanagawa 229-8558, Japan
takuya@sw.it.aoyama.ac.jp, {kaz, duerst}@it.aoyama.ac.jp

また，今回はエンコードのみを目的とするライブラリを実装するため，NKF のエンコード指定以外のコマンド機能 [3] は継承しない．

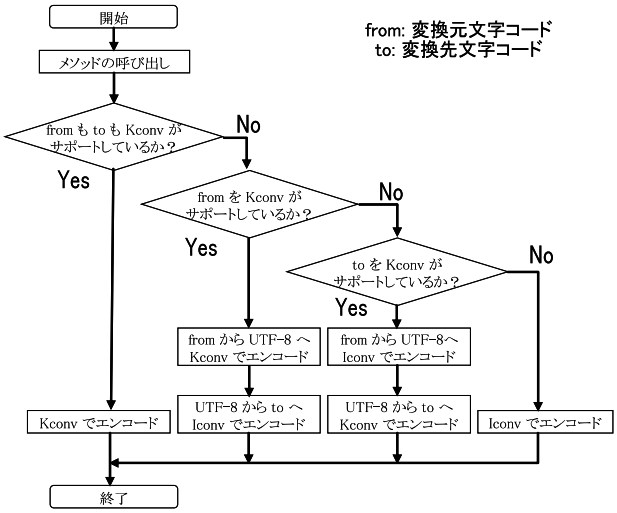

図 1: 処理のフローチャート

3.2 String クラスに追加されるメソッド

3.2.1 基本的な記述方式

基本的には Kconv と似た記述方式だが，変換元の文字コードと変換先の文字コードの引数指定の順序を逆にした．これは，3.2.2 節で述べるメソッドと順序を揃えるためである．from を変換元の文字コード，to を変換先の文字コードとして，以下にその記述の仕方を示す．

str.sconv(from,to)
例) *str*.sconv('UTF-8','CP1133')

3.2.2 ライブラリの名前を使わないメソッド

必要な処理としてユーザが求めているのは，特定の A という文字コードから特定の B という文字コードに変換することが多い．ライブラリの名前（例えば kconv）と同名のメソッドを使用することは省略できる．文字コード変換にライブラリを用いることは単なる仕組みであり，ユーザからはそれが見えないようにすることは Ruby の直感性に沿ったものであると考える．

本論文では，"A から B へ変換する" という単純なメッセージをオブジェクトに送るメソッドを提案する．また，Ruby の習慣に従い，メソッドの最後に ！ を付けると文字列が上書きされる仕様にした．以下に記述の仕方を示す．

str.A_to_B
例) *str*.SJIS_to_UTF8

str.A_to_B!
例) *str*.JIS_to_EUC!

しかし，Iconv がサポートしている文字エンコーディング名の数は 1000 を超える（一部重複して同じ文字エンコーディングを表しているものもある）．そのすべてについての組み合わせを用意するのは現実的ではない．よって，Ruby の伝統から，日本語を表現する文字コード群に対してのみあらかじめ用意することにした．この方式のメソッドがサポートしているのは，日本語で主に使われている ISO-2022-JP（JIS），EUC-JP（EUC），Shift_JIS（SJIS），UTF-8（UTF8），UTF-16（UTF16）の 5 つの文字エンコーディング間の相互変換のみである．

3.2.3 日本語文字コード以外への対応

3.2.2 節で述べたメソッドで日本語以外の文字コードを扱うために，メソッドを動的に作成する仕組みを用意した．以下に例を示す．

str.CP1258_to_CP1133

このように書いた場合，もしまだこのメソッドが定義されておらず，この変換が iconv でサポートされているものであれば，メソッドを動的に作成し，*str* をエンコードする．その後，.CP1258_to_CP1133 というメソッドが使えるようになる．

4 まとめ

本稿では，Ruby において標準添付されている文字コード変換ライブラリ群を画一的に利用できるライブラリを実装した．このライブラリを提案する意図は，将来的に，Ruby における文字エンコーディングの相互変換が一意的な標準機能として提供されることを望むものである．

文字コード変換は国際化の一つの基本となる要素だが，これからの Ruby の多言語処理環境構築において，世界の文字や言語に対する様々な機能を提供することが課題である．

参考文献

[1] Tate, B.: *From Java to Ruby*, Pragmatic Bookshelf (2006).

[2] 松本行弘, 綱手雅彦：スクリプト言語 Ruby の拡張可能な多言語テキスト処理の実装, 情報処理学会論文誌, Vol. 46, No. 11, pp. 2633–2642 (2005).

[3] Editors, R.: Rubyist Magazine 標準添付ライブラリ紹介【第 3 回】Kconv/NKF/Iconv, http://jp.rubyist.net/magazine/?0009-BundledLibraries.

[4] libiconv, http://www.gnu.org/software/libiconv/.

[5] CORPORATION, M. L.: Samba 国際化プロジェクト：ミラクル・リナックス, http://www.miraclelinux.com/technet/samba30/iconv_issues.html.