6

Muleを捨てて，Emacsを使おう
ー Emacsの自然言語処理機能ー

高橋 直人
電子技術総合研究所知能情報部
ntakahas@etl.go.jp

錦見 美貴子
電子技術総合研究所情報科学部
nisikimi@etl.go.jp

半田 剣一
電子技術総合研究所知能情報部
handa@etl.go.jp

戸村　哲
電子技術総合研究所情報アーキテクチャ部
tomura@etl.go.jp

本稿では，まず最初にMuleとGNU Emacsの発展の歴史について簡単にまとめた上で，現時点におけるGNU Emacs最新版の特長を述べる．次に新しい漢字入力モジュールTamago Ver.4の紹介を行う．またGNU Emacsを使って自然言語処理を行おうとしている研究者のために，Emacs Lispによる文字列処理プログラムの例を示し，解説を加える．最後にGNU Emacsの未来像について述べる．

まだMuleなんか使っているんですか？

Free Software FoundationがGNU Emacs（グニューイーマックス）[1]は多機能エディタと呼ばれることが多いが，実はエディタというよりも，一種のユーザ環境と呼ぶ方がふさわしい．一般的な文書の編集以外にも，デバッガの起動からソースプログラムの修正まで含めたプログラムの開発，電子メールやネットニュースの読み書き，その他数多くの操作がすべてGNU Emacsの中から行える．

このGNU Emacsに日本語処理機能を追加したNemacs（エヌイーマックス）が電子技術総合研究所から公開されたのは1987年6月のことであった．Nemacsは英語と日本語のみに対応した，いわゆるローカライズされたソフトウェアであった．Nemacsは1990年6月のNemacs-3.3.2（藤娘バージョン）を最後に開発が終了し，以降はMule（ミュール）がこれにとって代わった．

公開時期	バージョン	ベースシステム
1987年 6月	Nemacs-1.1	Emacs-18.47
1990年 6月	Nemacs-3.3.2	Emacs-18.51
1993年 8月	Mule-1.0	Emacs-18.59
1994年 2月	Mule-1.1	Emacs-18.59
1994年 8月	Mule-2.0	Emacs-19.25
1994年11月	Mule-2.1	Emacs-19.27
1994年12月	Mule-2.2	Emacs-19.28
1995年 7月	Mule-2.3	Emacs-19.28
1997年 9月	Emacs-20.1	
2000年 6月	Emacs-20.7	
?	Emacs-21	

表-1　Nemacs/Mule/Emacsの開発の流れ

Mule（Multilingual Enhancement to GNU Emacs）は英語と日本語だけでなく，その他の多くの言語も同時に扱うことができる，いわゆる国際化されたソフトウェアであった．Muleの開発は1991年に始まり，1992年5月のベータ版公開を経て1993月8月にはMule-1.0（桐壷バージョン）が発表された．その後も順調にバージョンアップを重ね，1995年7月にはGNU Emacs-19.28をベースにしたMule-2.3（末摘花バージョン）が公開された．電子技術総合研究所から公開されたMuleとしては今のところこれが最後となっている☆1．

その後Mule-2.3の多言語機能を本家のGNU Emacsに取

☆1 その後有志によって，GNU Emacs-19.34をベースにしたMule-2.3が公開されている．

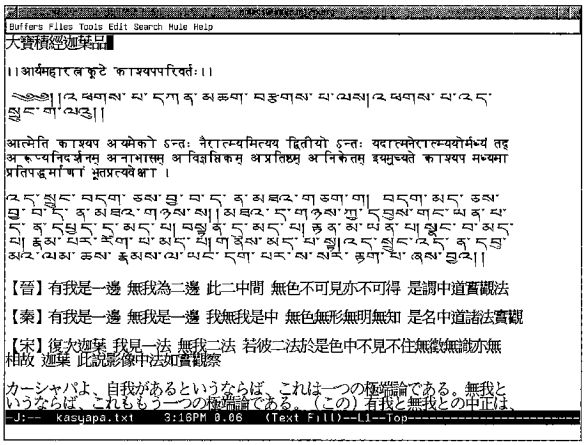

図-1　GNU Emacsによる多言語表示の例

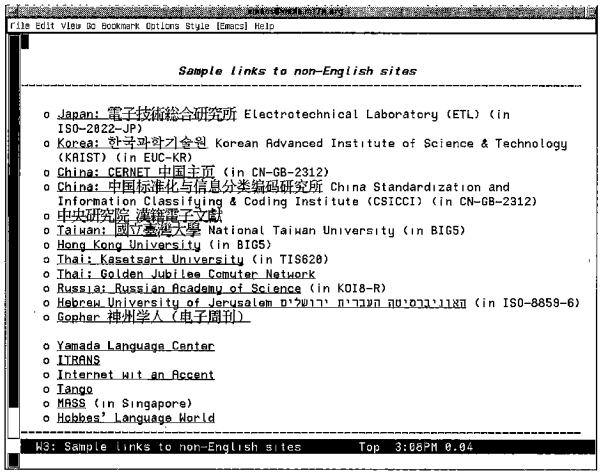

図-2　GNU EmacsをWebブラウザとして使う

り入れる作業が開始された．その最初の成果が1997年9月に公開されたGNU Emacs-20.1である．これによってGNU EmacsとMuleとの統合が完了し，これ以降の開発はGNU Emacsという統一された名前の下で行われることになった．以上をまとめると表-1のようになる．

本稿執筆時（2000年9月）での最新版は2000年6月に公開されたGNU Emacs-20.7☆2である．GNU Emacs-20.7はGPL（GNU General Public Lincense）第2版に従うフリーソフトウェアであり，商用，フリーを問わずほとんどのUNIX系OSで動作する．また旧DEC社（現Compaq社）のOSであるVMSや，MS Windows上でも動作する．図-1にGNU Emacs-20.7による多言語表示の例を，また図-2にGNU EmacsをWebブラウザとして利用している例をそれぞれ示す．

1995年からメンテナンスされていないにもかかわらず，Mule-2.3は現在でも雑誌の付録CD-ROMに収録されることがある．また現行の多くのLinux配布パッケージもMule-2.3を含んでいる．さらにMule-2.3の使い方を解説した書籍の出版も続いている．このため日本国内においてはいまだにMule-2.3を使い続けているユーザも多い．これはMule-2.3が非常に安定したソフトウェアであることを示しているが，やはり最新版であるGNU Emacs-20.7に比べると機能的に見劣りがすることは否めない．後述するようにEmacs Lispを使うことで自由に機能を拡張できることがGNU Emacsを始めEmacs一族の特長であるが，新しく書かれた便利なEmacs Lispプログラムも Mule-2.3では動作しないことがある．Mule-2.3の多言語機能のうちGNU Emacs-20.7で利用できないのはbi-directionality（左から右に書くテキストと右から左に書くテキストを混在させる機能）くらいである．そういった意味で，アラビア語やヘブライ語のように右から左に書く言語を扱いたいという要求でもない限り，すべてのユーザは今すぐMule-2.3を捨ててGNU Emacs-20.7に乗り換えるべきである．

☆2 ftp://ftp.gnu.org/gnu/emacs/emacs-20.7.tar.gz およびftp://ftp.gnu.org/gnu/emacs/leim-20.7.tar.gz

名称	URL
SKK	<http://skk.kuis.kyoto-u.ac.jp/index-j.html>
T-Code	<http://openlab.ring.gr.jp/tcode/index.html>
TUT-code	<http://www.crew.sfc.keio.ac.jp/~chk/intro.html>
yc.el	<http://www.ceres.dti.ne.jp/~knak/yc.html>
emcws	<ftp://ftp.ki.nu/pub/emcws/>

表-2　GNU Emacs-20用の各種日本語入力ソフトウェア

Emacs上の多言語入力環境Tamago Ver.4

「たまご」とは

「たまご」とは「たくさん またせて ごめんなさい」の略である．英訳してeggと呼ばれることもある．最初の「たまご」はNemacsの日本語入力メソッドとして開発され，外部変換サーバとしてはWnn（うんぬ）のjserver（ジェーサーバ）を利用していた．

NemacsがMuleに引き継がれて多言語化されると，「たまご」も日本語だけでなく中国語簡体字および韓国語をサポートするようになった．中国語変換サーバとしてはcserver（シーサーバ），韓国語変換サーバとしてはkserver（ケーサーバ）が利用された．また日本語変換サーバとしてはjserver以外にもsj3serv（エスジェースリーサーブ）が利用できるようになった．さらに，「たまご」とは独立した入力メソッドとして，cannaserver（カンナサーバ）を利用した日本語入力も可能となった．

Tamago Ver.4の特長

Emacs-20.5以降ではまったく新しい「たまご」であるTamago Ver.4☆3が利用可能である．Tamago Ver.4の主な特長としては，

- 完全Emacs Lisp化
- 多言語機能の強化

が挙げられる．

まず第1の特長である完全Emacs Lisp化について説明する．Mule時代の「たまご」は，外部変換サーバとの通信にCで書かれたライブラリを利用していた．したがって新しい変換サーバを利用するためには，Mule自体の再コンパイルが必要であった．またjserverとsj3servを同時に使うことができなかったため，「日本語入力にはsj3servを使いつつ中国語入力にはcserverを使う」などということは不可能であった．さらに，もしCライブラリ中でSIGSEGV等のシグナルが発生した場合，最悪Mule自体がクラッシュする可能性があった．Tamago Ver.4では，従来Cライブラリを用いていた部分を含めすべてをEmacs Lispで記述することにより，上記の問題を解決した．

次に第2の特長である多言語機能の強化について述べる．元来「たまご」は日本語入力メソッドとして開発されたため，中国語あるいは韓国語用の入力メソッドとしては機能的に不足している部分があった．特に問題となっていた点としては，

- 中国語繁体字が入力できない
- 中国語，韓国語入力への切り換えが面倒
- 中国語，韓国語ではいわゆる「全角」英数字が入力できない

などが挙げられる．Tamago Ver.4では外部変換サーバとしてtserver（ティーサーバ）を用いることで中国語繁体字入力を可能とした．また入力モードを充実させ，さらに各モード間の切り換えを改善することで上記の問題を解決した．

さらにひらがなとピンインを並べて入力したあと，全体をまとめて漢字に変換するようなことも可能になった．もちろんこの場合，ひらがなの部分はjserverを使って日本語漢字に変換され，ピンインの部分はcserverあるいはtserverを使って中国語漢字に変換される．ここでは詳しくは述べないが，ほかにもマルチフェンス，再変換，逆変換などさまざまな新機能が実現されている．

Tamago Ver.4から利用可能な外部変換サーバしては，Free Wnn（フリーうんぬ）☆4のjserver, cserver, tserverの3種類，Wnn6（うんぬシックス）☆5のjserver，そしてSJ3（Ver.1およびVer.2）のsj3servが挙げられる．

なお，現在ではEmacs-20用の日本語入力手段としてTamago Ver.4以外にも**表-2**のようにさまざまなソフトウェアが有志によって公開されている．興味のある

☆3 http://www.m17n.org/tamago/
☆4 http://www.freewnn.org/
☆5 http://www.omronsoft.co.jp/SP/unix/wnn6/ およびhttp://www.omronsoft.co.jp/SP/pcunix/wnn/

1.160 位置・地点・場合

1 *位置 *場面 シーン *場（ば） *立場 境 *境遇
境地 境涯 順境 逆境 佳境 悲境 苦境 進境
心境 危地 窮地 死地
2 *地位 位（い，くらい） 地歩 高位 低位 同位
帝位 王位 王座 官位 栄位 *首位 *上位 下位
優位 席次 上席 上座（じょうざ）
上座（かみざ） 下座（しもざ）
3 配置 部署 ポスト シート
4 *場合 *段階 局面 大局 時局 政局 戦局
5 破目 行きがかり 急場 難局 破局

1.1610 時間

1 *時（とき） *時間 タイム
2 *年月（ねんげつ） 年月（としつき） 歳月
月日（つきひ） 日月（じつげつ） 時日 光陰
星霜 *春秋

図-3 分類語彙表（フロッピー版）の一部

```
(defun bunrui-example (filename)                                          ;  1
  "Extract information from the bunrui-goihyou file FILENAME."            ;  2
  (interactive "FBunrui goihyou: ")                                       ;  3
  (let (ibuffer obuffer hyouki yomi)                                      ;  4
    (setq ibuffer (generate-new-buffer "*bunrui input*")                  ;  5
          obuffer (generate-new-buffer "*bunrui output*"))                ;  6
    (set-buffer ibuffer)                                                  ;  7
    (insert-file-contents filename)                                       ;  8
    (while (re-search-forward "\\(\\cj+\\)\(\\(\\cj+\\)\)" nil t)         ;  9
      (setq hyouki (match-string 1)                                       ; 10
            yomi (match-string 2))                                        ; 11
      (save-excursion                                                     ; 12
        (set-buffer obuffer)                                              ; 13
        (insert hyouki " " yomi "\n")))                                   ; 14
    (switch-to-buffer obuffer)))                                          ; 15
```

リスト-1 Emacs Lispプログラムの例

読者は比較してみるのもいいだろう．

自然言語研究にEmacs Lispを使おう

よく知られているように，Emacsの高次機能はEmacs Lisp[2]で書かれている．Emacs Lispはその名前が示すとおりLispの一種であり，しかもエディタの機能を拡張することを目的に設計されている．したがって，テキストファイルの編集に要求される複雑なパターンマッチングや置換などのきわめて高度な文字列処理が簡単に行える．このことはEmacs Lispが自然言語処理に適した処理系であることを示す．

本章ではEmacs Lispによる文字列処理の例として，国立国語研究所の日本語シソーラスである分類語彙表[3]（フロッピー版）から必要な情報を取り出すプログラムを示し，それに説明を加える．

図-3に分類語彙表（フロッピー版）の一部を示す．「1.160」あるいは「1.1610」と書かれている部分が分類番号である．リスト-1に示すプログラムは，パターンマッチを利用して後ろに括弧のついた語のみをこのデータ中から取り出し，

表記　読み
表記　読み
...

という形に整形する．

Emacs Lispでは，バッファという概念が重要な役割を果たす．バッファ（buffer）は，一種の作業領域であり，文字を挿入したり削除したりすることができる．

Emacsは同時に複数のバッファを持つことができる．また，バッファ内の作業位置はポイント（point）と呼ばれる．

それではプログラムの説明に入ろう．このプログラムは，bunrui-exampleという関数を定義する．この関数は，2つのバッファを用い，1つのバッファに分類語彙表のデータを読み込み，特定のパターンの文字列をそこから切り出してもう1つのバッファに書き込む．1，2行目から，この関数は，filenameを仮引数として持つこと，分類語彙表から情報を取り出すものであることが分かる．

3行目はEmacs Lisp特有のinteractive関数であり，bunrui-exampleが対話的に起動された際に仮引数filenameに何をセットするかを規定する．ここでの引数 "FBunrui goihyou: " の持つ意味は以下のとおりである．

1. 最初の文字Fはミニバッファを使ってファイル名を受け取り，それを仮引数filenameにセットすることを示す．
2. 続く文字列 "Bunrui goihyou: " はファイル名を受け取るときのプロンプトを示す．

したがってユーザがキーボードから

```
M-x bunrui-example RET
```

と打ってこの関数を対話的に起動すると，Emacsはミニバッファに

```
Bunrui goihyou:
```

というプロンプトを出してファイル名の入力を促し，ユーザが入力したファイル名を仮引数のfilenameにセットする．またこのとき入力されるものがファイル名であると分かっているので，ユーザはスペースキーやタブキーを使ってファイル名の補完をすることができる．このようにEmacs Lispでは，interactive関数を用いることで対話的プログラムも容易に記述できる．

5～8行目は，切出し作業の準備である．具体的には，作業領域として用いる2つのバッファを作り，そのうちの1つibufferを編集の対象として設定し，引数filenameとして与えられたファイルの内容を編集対象バッファのポイントの後ろに挿入する．この結果，ibufferには分類語彙表の内容が入り，ポイントはバッファの先頭に位置する．

次いで9～14行で，プログラムの中心である文字列の切り出しを行っている．具体的には，9行目に含まれる正規表現をibufferの中で検索し，それを整形してもう1つのバッファobufferに挿入する．この作業を正規表現が見つからなくなるまで繰り返す．文字列検索に成功すると，バッファ内のポイントは検索された文字列の直後に移動する．このため，次々と新しい文字列が検索できる．

9行目の正規表現は，文字列中では "\\" が "\" を，"\(" と "\)" がそれぞれ "(" と ")" と表記されることから，以下の4つの部分に分けられることが分かる．

1. \cj+を \(と \) で囲んだ部分
   "\\(\\cj+\\)"
2. 開き括弧1文字 "\ ("
3. \cj+を \(と \) で囲んだ部分
   "\\(\\cj+\\)"
4. 閉じ括弧1文字 "\)"

\cjはEmacsによる正規表現の拡張であり，任意の日本語1文字にマッチする．\(と \) は一度マッチした文字列を後で参照する際に用いられるメタ記号であり，検索時のマッチングパターンには含まれない．結局この正規表現は「日本語1文字以上の繰り返し，開き括弧，日本語1文字以上の繰り返し，閉じ括弧」というパターン，すなわち「表記（読み）」というパターンを表している．このように，拡張された正規表現によって複雑なパターンを容易に記述し，パターンマッチを要する処理を柔軟に行うことができる．

10～14行目では，いま見つかったパターンの一部をobufferに挿入している．10，11行目のmatch-stringは，前回の検索でマッチした文字列から参照用のメタ記号に囲まれた部分を返す関数である．ここでは，それぞれ表記と読みが返され，12行目以降でそれらがもう1つのバッファobufferに挿入される．

最後の15行目の関数switch-to-bufferは，引数として与えられたバッファを画面上に表示するものであり，これによってユーザはプログラムの実行結果を見ることができるようになる．

以上のプログラムに適当なファイル名，たとえばbunrui.el（Emacs Lispのソースファイルには.elという拡張子をつける習慣がある）をつけてセーブし，

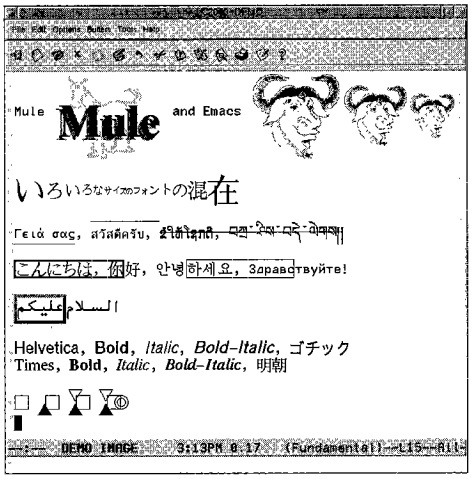

図-4　GNU Emacs-21の画面例

Emacsから

```
M-x load-file RET bunrui.el RET
```

とすればプログラムがロードされる．その後

```
M-x bunrui-example RET
```

とタイプするとミニバッファに

```
Bunrui goihyou:
```

とプロンプトが出るので，これに対して分類語彙表の入ったファイル名を打ち込めば

```
*bunrui output*
```

というバッファに入った処理結果が表示される．

このほかにもEmacs Lispには，高度な文字列処理に必要とされるさまざまな機能が提供されている．また必要に応じて外部プログラムと同期的あるいは非同期的に通信することも可能である．日本語形態素解析プログラムや翻訳ソフトウェアなどと組み合わせれば，複雑な自然言語処理も比較的簡単に実現できよう．実際にEmacs Lispでプログラムを書いてみようと思われる方には，GNU Emacsに付属しているEmacs Lispのソースプログラムをオンラインマニュアルを活用しつつ読んでみることをお勧めする．

Emacsはどこへ行く？

本稿執筆時点で最新のGNU Emacsのバージョンは20.7であるが，現在開発中の新バージョンはGNU Emacs-21になると計画されている．本章では，そのGNU Emacs-21に搭載される予定の機能について説明する．

GNU Emacs-21における最大の変化は新しいディスプレイエンジンが導入されることである．この結果として，ユーザインタフェース全体がよりグラフィック指向のものとなる．

図-4に，新たなディスプレイエンジンを用いた画面の様子を示す．この図で分かるように，文字の幅や高さが可変になり，また文字列に対してさまざまな属性（attribute）を付加することができるようになる．付加できる属性には，下線，上線，見え消し，2次元・3次元boxなどが含まれる．バッファ内にイメージを含めることもできるようになる．

さらに，バルーンヘルプに類似したユーザ支援のシステム，点滅カーソル，行折り返し表示の一層のビジュアル化（画面の両端にfringeと呼ばれる領域が確保され，そこに折り返し矢印が表示される）などが新たにサポートされる．

複数のグリフイメージ（画面上に現れる具体的な画像としての文字）を組み合わせて文字を表示する手法に関しても大幅な変更が予定されている．現行のGNU Emacs-20でグリフイメージを組み合わせる場合は，合成文字という特別な文字の単位を使用する．これに対し，GNU Emacs-21では文字列に対するpropertyとして組み合わせを指定する．こうするとバッファの中では合成前の各要素文字がそのまま見えるので，サーチなどが便利になる．

グラフィック以外の機能としては，音声データを一般の文字列と同様に扱ったり再生することができるようになること，X Window Systemの標準プロトコルであるXIMの正式サポートによりkinput2やATOKなどの日本語入力プログラムとのインタフェースが自然にとれるようになることが予定されている．また，新しいデフォルトの文字集合として，JISX0213-1, JISX0213-2, Unicodeの一部（U+0100～U+24ff），その他が含まれる予定である．

またEmacs-21.2からは，Mule-2.3よりも高度な機能を持つbi-directionalityがサポートされる予定である．このようにGNU Emacsは着々と進化を続けている．だからMuleを捨てて，Emacsを使おう．

参考文献
1) リチャード・ストールマン: GNU Emacsマニュアル20.6, アスキー, 東京 (2000).
2) ビル・ルイス他: Emacs Lispリファレンスマニュアル, アスキー, 東京 (2000).
3) 国立国語研究所編: 分類語彙表, 東京 (1964). (フロッピー版は品切れ)

(平成12年9月25日受付)