

視覚障害者の コンピュータ利用と 日英2カ国語 音声化システム BEP

渡辺 隆行

東京女子大学・現代文化学部・コミュニケーション学科
nabe@twcu.ac.jp



視覚障害者は、音声合成で文字を読み上げることでパソコンを使用している。視覚を使わずにコンピュータを使いこなしている彼らの音声利用の現状を知ることによって、実用的な音声出力システムに必要な新しい手がかりを得ることができるかもしれない。本稿では、筆者が視覚障害者と共同開発した日英2カ国語音声化システム BEP（Bilingual Emacspeak Platform）を題材に、視覚障害者の音声によるコンピュータ利用の現状と問題点を述べる。



音声でのパソコン利用と視覚障害者

人間同士の情報交換では音声対話が大きな役割を果たしており、情報処理システムとの情報交換も音声対話で行うことが人間にとって自然なインタラクションである。しかし現実には、音声によるコンピュータ利用は限られた場面でしか使われていない。コンピュータと人間の音声対話には、人間同士の音声対話と比べて欠けている機能が多い。

全盲の視覚障害者は、入力にはキーボードを使うけれども、音声合成機能を用いて文字を読み上げさせてパソコンを使用している^{☆1}。彼らの利用は音声対話ではなく、コンピュータが持っている情報を聴覚を使って取得しようというものであるが、視覚を使わずにコンピュータを使いこなしている彼らの音声利用の現状を知ることによって、実用的な音声出力システムに必要な新しい手がかりを得ることができるかもしれない。

☆1 点字には音声にはない利点があるので、点字を読める視覚障害者は音声出力のほかに点字出力も併用していることが多い。



視覚障害者の音声利用の特徴

■音声出力の特徴

音声出力は、視覚を使った情報出力に比べて以下の特徴を持つ。

- 眺望感がない。望む情報がいつ出力されるのかを最初にざっと眺めて知ることができず、その情報が出力されるまで待たなければいけない。
- 再生された音は消えてしまい、聞き返すことができない。
- 目で読むよりも耳で聞くほうが約2倍の時間がかかる。
- 音は拡散する。
- 視覚が使えない状況でも情報を伝達できる。
- ながら作業ができる。ほかの作業をしながら音声に耳を傾けて情報を取得することが可能である。

このような特徴を持つ音声出力でパソコンを効率よく使うために、視覚障害者用音声化システムには上記の欠点を補うような工夫が施されている^{1)～3)}。

「朝日が昇る」

- (1) 詳細読み ; 朝 アサノ チョー, 麻 アサノ マ, 浅 アサイノ セン
- (2) 簡易読み ; チョー ニチ ガ ショー ル
- (3) 滑らか読み ; アサヒガノボル



図-1 漢字音声化の工夫

■音声出力における文字種の区別などの工夫

視覚障害者の場合、コンピュータにある文字情報を正確に取得するという目的を持っており、一般の利用者がコンピュータの利用に音声対話を用いるのとは音声出力に求める条件が少し異なる。すなわち、音声対話の場合はもともと音声情報であったり音声情報として生成されたりした情報をユーザに伝えるのに対し、視覚障害者が音声出力を用いるのは、コンピュータが持っている文字情報を正確に効率的に聴覚を通して取得したいという目的がある。したがって視覚障害者の場合、大文字や小文字などの文字の種別や句読点がどこにあるかなども知りたい場合がある⁴⁾。

文字種を英語だけに限っても、句読点や特殊記号を読むかどうか、あるいはどの程度まで記号類を読むかなどの punctuation mode を指定できる必要がある。また空白文字を指摘できる機能、短い言葉やビープ音や声の高低や種類を変えることなどで大文字を小文字と区別して読み上げる機能、文末で少し間を置いたり短い音を出したりする機能も必要である。その他にも単語として読むだけでなくスペルアウトして読み上げる機能、alpha などのようにフォネティックコードでアルファベットを読む機能、同じ記号が連続しているときは数に変換して読み上げる機能なども必要である。

日本語にはさらに多くの機能が必要となる (図-1)。

- ・同音異義語が多い漢字の読み上げには3種類のモードが必要である。

- (1) 詳細読み ; 漢字を音声で識別するために「タンボの田」などのユニークな説明つきで読み上げる機能。カナ漢字変換時に必須。
- (2) 簡易読み ; カーソルを移動する際の一文字読み上げ時などにできるだけ簡単に漢字を音声表示する機能。音読み (や訓読み) をする。
- (3) 滑らか読み ; 文脈の中で構文解析をして正しく読む機能。現在は音声合成ライブラリや音声合成装置がこの機能を担ってくれるので視覚障害者用音声化システムとしては特に仕事をしなくてよい。

- ・カタカナを音声で識別する機能も必須である。声のピッチまたは男女を変えてカタカナを識別している場合が多い。
- ・ローマ字、数字、カタカナの場合、全角と半角を区別する必要がある。声の高さで識別するなどの工夫がなされているものもあるが、まだ一般的ではない。
- ・ローマ字はアルファベットで書かれているが英語の音声合成では正しい発音にならないので正しい発音に変換する必要がある。Windows で使われている音声合成ライブラリなどにはこの機能がないので、音声出力システムの中であらかじめ正しい発音のカタカナ文字に変換した上で音声合成ライブラリに読み上げさせる必要がある。
- ・日本人であっても英語を扱う機会が多い。日英2カ国語が混在した情報をそれぞれの言語の発音で正しく読み上げることができる実用的な音声化システムは我々が開発した BEP 以外にはない^{☆2}。既存の日本語音声化システムの中には、カタカナで発音を表記してある辞書を持っていて、英単語に出会ったらその辞書でカタカナ読みに変換してから日本語の音声合成ライブラリでジャパニーズ英語風に音声化することができるものが多い。



視覚障害者用音声化システムの問題点

■ OS の問題

パソコン用音声化システムが登場した当時の OS は MS-DOS であり、DOS は文字でコマンドを入力し文字で情報が出力されるので音声化しやすかった。1990 年代に入ると初心者にも使いやすい GUI (Graphical User Interface) を持った Windows が登場した。Windows は最新の技術がいち早く導入される反面、ウインドウ・アイコン・メニュー・ポインタなどの GUI 部品による操作

^{☆2} 英単語を英語の合成音声で読み上げる辞書検索ソフトのような、狭い範囲でバイリンガルなソフトは存在する。

2002年5月

日	月	火	水	木	金	土
28	29	30	1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	

図-2 カレンダーの表示

を前提にしている場合が多いので、Windows 標準の部品を使ったりスクリーンリーダに配慮して作成したりしないと視覚障害者が使えないアプリケーションになってしまう。またバージョンアップの周期が早い巨大な OS に合わせてスクリーンリーダを開発するのは困難な作業である。そのため、Windows は使いにくいと感じる人がいたり、新しいバージョンでは使えない機能が生じたりする。Windows でなくてもできる作業ならば DOS を使う視覚障害者も存在する。

Windows 以外の高機能な OS として、Linux などの UNIX 系 OS が注目を集めている。UNIX はもともとテキストコンソールから操作する OS であり、文字情報だけで OS を管理できる。したがって、UNIX も DOS と同じように音声利用できて DOS を利用している視覚障害者が移行しやすい OS であるはずであるが、ユーザが少ないこともあって UNIX 用のスクリーンリーダは数少なく、日本語スクリーンリーダで実用化されているものがない。

■アプリケーションの問題

Windows では、Internet Explorer や Office アプリケーションなども音声で利用できるようになってきた。しかしまだまだ音声で利用できないアプリケーションが多く存在する。

■情報取得の仕組みが不足している問題

Windows を例にとると、Windows にはもともと画面表示専用で作られた OS であるので、初期の音声化システムは OS が画面に表示する情報を音声化するしかなかった。その後 Windows にも MSAA (Microsoft Active Accessibility interface) という仕組みが組み込まれ、一部の情報は MSAA を通じて自由に取得できるようになったが、MSAA に対応していないアプリケーションはこの方法では音声利用できない。Java の Swing では、自由に情報にアクセスする機能が組み込まれている。このよう

に、システムが持っている情報のレイヤーとユーザに表示するレイヤーの間に、任意のデバイスに表示できるような情報取得の仕組みが必要である。

■データ構造とユーザインタフェースの問題

画面表示とは別の音声表示専用ユーザインタフェースの必要性を示す好例がカレンダーの表示である。カレンダーは 2 次元の表であり、晴眼者は目で見ただけで今日が何曜日かわかる (図-2)。しかしカレンダーを単純に左から右へと行に沿って読み上げるだけ、あるいはカーソルがある現在の日付を読み上げるだけではこのような情報は得られない。カレンダーを音声で表示する際は、「2002 年 5 月 1 日、水曜日」のようにカレンダーから必要な情報を取得した上で音声化する必要がある。またこのような情報の加工を容易にするためには、カレンダーという構造を持った形でデータを表現しておくことが望ましい。

■正確な音声化ができない問題

職場での仕事には正確さと速度を求められるので、仕事で使う音声化システムは情報を効率よく正確に音声化しなければならない。ここで言う正確さには、文字種を聞き分けられることやローマ字や英語を正しい発音で読み上げること含まれる。しかし前章でまとめたように、全角と半角の区別、ローマ字や英語の音声化にはまだ不十分な点がある。

■音声合成の表現力の問題

Web ページは HTML でデータをマークアップしているので、各要素を適切に音声化しやすいはずである。また CSS2 (Cascading Style Sheets, Level2) ^{☆3} を使えば音声表示用のスタイルシートも定義できる。リンク要素を女性の声で読み上げるというのは、既存の音声化システムでもすでに実用化されている。しかし、表の見出し要素や定義つきリストの見出しの声をえたり、H1、H2 などの見出し要素の深さに応じて声の高さを変えたりといった工夫を実現している日本語音声化システムはまだない。これは、日本の音声合成機能の表現力が不足していることも一因である。

DECTalk という英語の音声合成装置は、9 つの声 (男性の声 4 種、女性の声 4 種、子供の声 1 種) を持ち、文字単位や単語単位や句単位で音声化でき、75wpm から

^{☆3} Cascading Style Sheets の詳細は、
<http://www.w3.org/TR/REC-CSS2> 参照。

画面表示にはいろいろな
フォントを使用できます。

```
int main(void) {  
    printf("Hello, world. %n");  
    return 0;  
}
```

色分けしたりインデントすることで
ソースコードが見やすくなります。

音声表示でも
いろいろな声 (Voice フォント)
を使うことができます。

異なる人の声や男女の声、
声の高さや太さやイントネーション
を変えるのです。

ただし音は1次元の表示です。

図-3 Voice フォントの概念

600wpm (約3倍速) の速度で明解な音声合成ができる。この他にも、コマンドを与えたり細かなパラメータを指定したりすることで平坦な声や特徴のある声などを作することもできる。Emacspeak という英語の音声化システムは、この DECTalk の豊かな表現力をフルに活用して HTML の音声化ができる。この Emacspeak は Voice フォントという概念 (図-3) を持っており、プログラムソースを編集するときはコメント、キーワード、関数名、文字列、などの構成要素ごとに異なった声を割り当てることにより、色つきの画面が晴眼者にとって使いやすのと同等な使いやすさを音声ユーザに提供している。つまり、GUI 画面に表示される文字に複数のフォントがあり太字やイタリックなどの修飾をしたり大きさを変えたりできるように、音声においても、声の主を変えたりピッチを変えたりイントネーションを平坦にしたり男女を変えたり年齢を変えたりすることで、テキストの読み上げに対して多様な修飾ができる。しかし DECTalk のような豊富な表現力を持つ既存の日本語音声合成はまだない。

■ 音声出力の情報伝達速度の問題

視覚障害者は音声合成の最高速度で音声出力システムを利用している。日本語音声合成ソフトウェアの最高速度は1.5倍から2倍程度なので読み上げにかかる時間を半分に短縮することができる。視覚障害者に聞いたところ、倍速程度なら使っているうちに耳が慣れて聞き取れるようになるそうである。3倍速で再生しても聞き取りやすい音声合成があれば、この時間をさらに短縮できる。視覚障害者のようにコンピュータが扱っている文字情報を取得することが目的である音声利用の場合、声は単なる文字符号のような役割に過ぎず、どの文字が使われているのかを正確に速く伝達することが重要になる。しかし今の音声合成の技術的な関心は人間のように自然に話

すことができる音声合成に集中しているので、高速でも聞きやすい音声合成の研究も望まれる。

伝達速度を早くするには、再生速度を速くする以外にも斜め読みをして大事なところだけを分かりやすく読み上げたりする方法も考えられ、このような方面の研究も行われている。



Bilingual Emacspeak Platform

■ BEP 開発の動機

日本にはすでに複数のスクリーンリーダーがある。これらはどれもよくできていて視覚障害者のかなりのニーズを満たしているけれども、前章で述べた問題: UNIX で使えない、仕事や研究で使えるアプリケーションが不十分である、正確な音声化が不十分である、英語を正しい発音で読み上げられない、音声に最適化したデータの表現がなされていない、声の種類が少なくて表現力が乏しい、もっと高速で効率よく読み上げさせたい、などの問題がある。

アメリカには、全盲の Raman 博士が、自分の博士論文執筆用に開発した音声化システム Emacspeak^{☆4}がある。Emacspeak は、UNIX で標準的に使われている高機能なテキストエディタ Emacs を、DECTalk などの表現力豊かな音声合成装置を用いて音声化するシステムである。

Emacspeak は普通のスクリーンリーダーとは異なり、Emacs エディタがスピーチサーバを制御するタイプの能動的な音声化システムであり、UNIX 上で動作する。

^{☆4} Raman が開発した Emacspeak は今も改良が続けられており、オープンソースとして <http://emacspeak.sourceforge.net/> で公開されている。

1. ネイティブ英語モード；
 - ・ right と light, but と bat を聞き分けることができる。
 - ・ エクスプローラと Explorer の違いもわかる。
2. 混在モード；
 - ・ いきなり英語で短いエラーメッセージを出されても聞き取れない。
 - ・ 長い英文はネイティブの発音で聞きたい。
3. カタカナ英語モード；
 - ・ プログラム言語のソースコードなどは英語で発音されると理解しにくい。
 - ・ 耳が慣れていなくてカタカナ英語を聞き取れない日本人はいない。

図-4 3種類のバイリンガルモード

Emacs は、扱う情報の構造 (Emacs のモード) によって動作を変えたり、Lisp パッケージで新しい機能を追加したりカスタマイズしたりできる。つまり Emacs は単なるエディタを超えたアプリケーションプラットフォームとも呼ぶべきソフトウェアであるので、これを音声化すると電子メール、Web、テキスト編集、プログラム開発、辞書検索などの、仕事や勉強の場で必要となるほとんどの機能を音声利用できるようになる。Emacspeak は以下の特徴を持つ。

- ・ AUI (Auditory User Interface) : 聴覚専用のユーザインタフェース⁵⁾ という考えを持っており、カレンダー表示やファイラーなどのモードでは、耳で聞いたときに分かりやすいようにカレンダーやファイラーを使用できる。
- ・ Voice フォント : コンテキストに応じて多彩な声を使い分けることができる。Emacs 用の Web ブラウザである W3 は ACSS (Auditory Cascading Style Sheets) に対応しているので、HTML の要素に応じて声を使い分けることもできる。
- ・ Auditory アイコン : 文字メッセージは再生に時間がかかるので、カーソル移動、ファイル消去、コンパイル終了などの各種イベントに対して短いサウンドを割り当て、イベントをユーザに迅速に通知する。バックグラウンドで動かしている作業の終了を auditory アイコンで知ることもできる。

Emacspeak は既存の日本語音声化システムにないユニークな特徴を持っており、これを日本語で利用したい日本の視覚障害者も存在した。そこで著者は、1999 年に晴眼者と視覚障害者の共同プロジェクトとしてバイリンガル Emacspeak プロジェクトを立ち上げた。このプロジェクトは、(1) 日英 2 カ国語を音声化できるソフトウェアスピーチサーバを Windows と Linux の 2 つの OS 用に開発、(2) Emacspeak を日本語と英語の 2 つの言語を扱えるように拡張し、既存の日本語音声化システムを参考

にして^{☆5} 日本語音声化に必要な機能を実装、(3) オープンソースとして公開しユーザのサポートを行う、ことを目標にしている。

■ 3種類のバイリンガルモード

BEP は日本語と英語が混ざった情報を読み上げる際に 3 種類の発音方法を使うことができる。(図-4)

- ・ ネイティブ英語モード : ASCII 文字はすべて英語と判定し、英語の音声合成を用いて読み上げる。1 文中に英語と日本語が混在していた場合も、その都度音声合成の言語を切り変えて正しい発音で読み上げる。
- ・ 混在モード : 一定の長さ (デフォルト 40 文字) 以下の ASCII 文字の連続は日本語と判定し、カタカナ読みで日本語の音声合成に送る。日本語文中に現れる短い英文字列は日本語化した語で日本語の一部として読み上げる方が自然と思われる場合が多いので、日本人用のバイリンガルシステムとして、このモードをデフォルトとしている。
- ・ すべてカタカナモード : ASCII 文字列もすべてカタカナ読みとして日本語の音声合成で出力する。プログラムなど特定の環境では、英数字のみで書かれた情報でも、すべてカタカナ英語で読み上げるほうが日本人にとって分かりやすい。

■ BEP の構成と処理の流れ

以下主に Linux 版の BEP を中心に述べる。図-5 に示すように、BEP は Emacspeak をバイリンガルに拡張した Emacs Lisp 部と、日本語と英語の音声合成エンジンを制御するスピーチサーバ部の 2 つに分かれる。スピーチサ

^{☆5} カタカナ英語の読み上げや漢字の詳細読みには、静岡県立大学の石川准教授が開発した音声化システム「グラスルーツ」で使われている辞書を使っている。

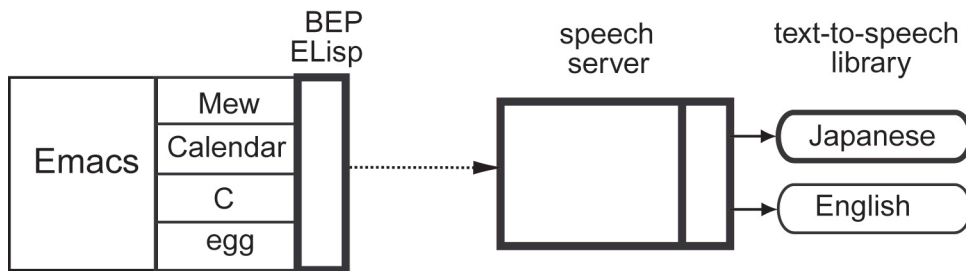


図-5 BEP の構成図

ーバは Emacs とは独立したアプリケーションで、Emacs のサブプロセスとして起動される。図中の太線枠は自主開発部を示す。Emacs Lisp 部は両 OS で共通しているが、スピーチサーバは OS ごとに開発する。

「これは Bilingual Emacspeak です」のように一行に英語と日本語が混ざっている場合、両言語の発音で読み分けるためには、日本語で「これは」と音声化した直後に英語で「Bilingual Emacspeak」と読み上げ、その音声化が終了した直後に日本語で「です」とタイミングを合わせて読まなければならない。同じ言語であっても声の種類を変えたい場合もある。このように声と言語の属性が異なる一連の文字列を連続して読み上げるために、属性の変わり目で切り分けた文字列を属性付きでキューに順番に格納してから、非同期でキューから順番に取り出して音声出力するようになっている。音声化停止コマンドが発行された場合は、キュー自体を破棄する。またカーソルを左右に移動させたときなどは移動先の文字を即座に読み上げてほしいので、このような場合は与えられた文字列を即座に読み上げる。

BEP の処理の流れを説明する。

I. Emacs Lisp 部：

- 言語属性および声質の付与：Emacs のバッファ中で読み上げ対象となり得るテキストについて、情報の種類に応じて声質の属性を割り当てる。声質の割り当て方はバッファのモードと結びついており、たとえば C モードだったら、コメント行は抑揚のない平坦な声になっている。声の属性を付与するときに言語属性も付与している。文字種（Emacs の文字カテゴリー）とその前後の文字列の文字種を考慮して、適切な言語属性をセットする。3 種類のバイリンガルモードによって ASCII 文字に対して貼り付ける言語属性は異なる。たとえば混在モードでは、読み上げ対象となっている文字列の前後の文字も調べて、ASCII 文字の長さが 40 文字以下だったらカタカナ英語の属性を与えている。バイリンガルモードはバッファローカルとするこ

ともでき、バッファのモードによって切り換えることも可能であるので、C モードのようにプログラムソースの編集を行うときはカタカナ英語モードに固定することもできる。

- Emacs Lisp の advice や hook などの機能を用いて、ユーザに伝えるべき情報を、Emacs の動作を妨げることなく Emacs から取り出す。たとえばカーソルが一行下に移動したら、移動先の行の内容を取り出す。
- 記号類を文字で置き換えたり、同じ記号が続くときに省略したり、不要な情報を読み上げないようにしたりするなどの処理を、取り出した文字列に施す。
- 読み上げ対象となっている文字列とその属性情報、即時停止などの要求をスピーチサーバに入力する。文字列の属性はここでインテキストコマンドなどの文字列に変換されて、スピーチサーバには単純な文字列が送られる。

II. スピーチサーバ部：

- 標準入力に送り込まれたコマンド付き文字列を解析する。即時停止、即時発声のコマンドはリアルタイムに処理される。また非同期な読み上げでは送られた文字列と属性情報をキューに送り込み、それを順に取り出して音声として出力する。
- 音声化の際は、Emacs Lisp で埋め込まれた言語属性に応じて動作が異なる。日本語属性のついた英数記号列はカタカナ英語辞書（英単語のキーに対し、カタカナで発音を書いてある辞書）を引いて英単語をカタカナに直してから日本語の音声合成ライブラリに送り込む。ローマ字を解析してカタカナに変換する作業もここで行う。

■ BEP の現状

Linux 版の BEP は 3 種類のバイリンガルモードを持ったシステムとして完成した。Windows 版は日本語と英語の自動切り換え、あるいは英語はカタカナ英語だけ、

という2種類のスピーチサーバができており、3種類同時に使えるように改良中である。両者とも自主開発部分はWeb (<http://www.argv.org/bep/>) でGPL (GNU General Public License) に従ったオープンソースとして公開している。このWebには音声や動画のデモなども用意されている。メーリングリストによるユーザサポートも行っている。

Linux版BEPを使うためには、本システム以外にEmacs 20.6以降、日本語音声合成エンジン(クリエートシステム開発(株)のDocument Talker for Linux)、英語音声合成エンジン(IBMのViaVoice)が必要である。Windows版を使うためには、Microsoft Speech API、宮下氏が開発したWindows版EmacsであるMeadow 1.14以上、日本語音声合成エンジン(バイリンガルモードによって、IBM ProTalker97 Ver. 2.04以上または東芝音声合成LaLaVoice)、英語音声合成エンジン(Microsoft Text-to-Speech Engine)が必要であり、これらのインストールに約100MBの空き領域が必要。詳細はWeb参照。



全盲の視覚障害者は音声出力や点字出力でパソコンを利用しているが、目で文字を読むことができない彼らが音声化システムに求めている機能は人間のように自然に音声対話することではなく、迅速に正確に分かりやすくコンピュータが持っている情報を音声化することである。現在の音声合成研究は自然な速度で発声したときにできるだけ人間的に聞こえることを第一の目標としているが、視覚障害者用の音声合成装置として必要なのは、2倍速や3倍速などの高速で発声させたときにも明瞭に認識できるような合成音声である。また、聴覚に最適化したユーザインタフェースも重要である。

日本人用の日英2カ国語音声化システムには、日本語と英語を正しく発音する機能と、ネイティブの発音とカタカナで表記した日本語風発音の2種類の英語発音方法が必要である。また、間違った発音で英語を読み上げるくらいならば、スペルアウトして読み上げたほうがよい。日本語音声化システムには、ローマ字を正しく読み上げる機能も必要である。

視覚障害者が必要としている機能を持った音声化システムをオープンソースで開発するために、Emacsが好きな晴眼者と視覚障害者が個人の資格で集まってバイリンガル音声化システムBEPを開発した。このシステムは既存の視覚障害者用日本語音声化システムおよびEmacspeakを基にしており、日英2カ国語が混在した情報を正確かつ迅速に音声だけでユーザに伝えることを目的としている。開発成果はWebで公開し、メー

ングリストなどを利用してユーザサポートも行っている。Emacsは初心者には敷居が高いアプリケーションであり、そのようなユーザに配慮した情報をまだ用意できていないこともあって現在のユーザ数はまだ10人程度であるが、英語の読み上げなどBEPにしかない機能に興味を持つユーザは多い。

一般にスクリーンリーダは高価なので、BEPのようにオープンソースでフリーな音声化システムの価値は高い。またオープンソースであるEmacspeakを基にBEPの開発が行われたように、ソースを自由に改変することで、ユーザのニーズに適したソフトウェアを開発しやすいという利点がある。

謝辞 釜江常好教授の紹介により著者はRaman博士のEmacspeakを知った。釜江教授はVoice Windowsという視覚障害者用のWindows利用支援ソフトの開発に取り組んでおり、BEPの開発も初期段階ではその一部であった。その後視覚障害を持つ井上浩一氏と出会ったことで、BEPの本当の開発がはじまった。誰よりもまず釜江教授と井上氏に感謝する。また、切明政憲氏、坂本貢氏、高橋玲子氏、南谷和範氏、吉本浩二氏、中村典嗣氏、そして高橋直人氏など、ユーザおよび開発者として貢献し有益な助言を授けてくれたBEPのメンバたちに感謝する。また、オープンソースとしてあるいは無償で開発成果を利用させていただいたT.V. Raman博士、G. Bishop教授、「グラスルーツ」の開発者である石川准教授に感謝する。本研究の一部は、日本学術振興会未来開拓学術研究推進事業における研究プロジェクト「高度マルチメディア応用システム構築のための先進的ネットワークアーキテクチャの研究」(JSPS-RFTF97R16301)による。また一部は情報処理振興事業協会の「平成13年度未踏ソフトウェア創造事業」による。ここに記して謝意を表す。

参考文献

- 1) 栗原 亨: 視覚障害者用音声端末の現状と将来, 東京大学大型計算機センター, CCUT-TR-89-01 (1990) .
- 2) 斎藤正夫: 視覚障害者支援ソフトウェアの製作, 情報処理, Vol.36, No.12, pp.1116-1121 (Dec. 1995) .
- 3) 石川 准: GUI用スクリーンリーダの現状と課題 - 北米と欧州の取り組みを中心に -, 情報処理, Vol.36, No.12, pp.1133-1139 (Dec.1995) .
- 4) 渡辺哲也: 視覚障害者のパソコン利用における音声による文字の区別, 職リハネットワーク, Vol.9/2, pp.28-31 (1996) .
- 5) Raman, T.V.: Auditory User Interfaces - Toward the Speaking Computer -, Kluwer Academic Publishers, Boston (1997) .

(平成14年5月11日受付)