

日本人用の日米2ヶ国語高機能音声化システム BEP

渡辺隆行^{a,b}、切明政憲^b

^a 湘南工科大学情報工学科 <takayuki@la.shonan-it.ac.jp>

^b ARGV (Accessibility Research Group for the Visually-impaired)

Bilingual Emacspeak Platform (BEP)とは、日本の視覚障害者が仕事や勉強や趣味の場で GNU Emacs を活用するための、日米2ヶ国語に対応した音声化システムである。ユーザのニーズを満たしたバイリンガル音声化システムを実現するための視覚障害者と晴眼者の共同プロジェクトとして、英語音声化システム Emacspeak のバイリンガル化と Windows 及び Linux 用の2種類のバイリンガルスピーチサーバの開発、BEP ユーザのサポートなどに取り組んでいる。本報告はまず、音声出力だけでパソコンを利用するために既存の日本の視覚障害者用音声化システムが備えている特徴をまとめる。次に、日米2ヶ国語が混じった情報をユーザにわかりやすく音声表示するための要件について BEP 開発の経験を元に議論する。

BEP: a practical bilingual speech synthesis system for Japanese

T. Watanabe^{a,b} and M. Kiriake^b

^a Department of Information Science, Shonan Institute of Technology,

^b ARGV (Accessibility Research Group for the Visually-impaired)

The Bilingual Emacspeak Platform (BEP) is a bilingual, Japanese and English, text-to-speech system for Japanese visually impaired Emacs users. Making full use of helpful functions of the existing Japanese screen-readers and trying to fulfill the requirements of advanced users, we are developing bilingual version of Raman's Emacspeak and bilingual speech servers for both Windows and Linux. In this paper, the authors firstly review the features of existing Japanese text-to-speech system and then, based on the experiences of BEP, show the requirements of a bilingual speech synthesis system for efficient and practical use.

1. はじめに

1.1 音声出力の可能性

初心者がコンピュータを使うためには、まずキーボードを習得しなければならない。情報の出力にディスプレイが使われている限り、ユーザはコンピュータの前に座り続けなければならない。しかし音声対話によってコンピュータと情報を交換できるよう

になれば、このような問題は解決される。このような背景から、人間のように感情豊かに発話する音声合成エンジンや、人間が普通に話す言葉を認識する音声認識エンジンの開発が精力的に進められており、音声対話によるコンピュータ利用が注目されている。

しかし現実には、音声対話は限られた場面でしか利用されていない。(日本語)音声認

識はまだ認識率が完全でないために修正に手間がかかるし、音声合成による情報出力は画面出力より時間もかかり集中して聞いていないと聞き逃しやすい。

ところが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、情報出力は音声合成だけでパソコンを使いこなしている。つまり、視覚をまったく使えなくても音声出力だけでコンピュータを使えることが視覚障害者たちによって証明されている。

1.2 日米2ヶ国語が混在した情報の音声化

普段われわれが処理する情報は日本語の中に英語が混在したり、日本語と英語の文や段落が混じったりしている。このような情報を目で読むときは、単語が半角 alphabets で書かれていようが全角 alphabets で書かれていようが、表現に使われている文字が異なっていることを明確に認知しながらも、意味としては特に意識することもなく同じものとして認識している。ファイル名やコマンド名などは、複数の英単語を連結してから短縮したり短文を短縮したりする方法で名付けることが多い。目で読むときは頭の中で適当に展開して、“cp”コマンドを“コピー”コマンドなどと認識して読んでいる。こういった情報を音声だけで伝達する場合、このような柔軟さは失われてしまい、何らかの工夫が必要になる。

2. 視覚障害者用音声化システムの現状

2.1 日本語音声化の工夫

音声出力は以下の特徴を持つ。

1. 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、望む情報が出力されるまで待たなければいけない。
2. 再生された音は消えてしまい、簡単には聞き返せない。
3. 目で読むよりも耳で聞くほうが時間が

かかる。

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いくつもの工夫が施されている[1-3]。MS-DOS 用の画面読みプログラムを開発した視覚障害者の斎藤正夫氏は、視覚障害者用音声化システムの留意点として以下の点をあげている[4]。

1. 読み上げ途中で不要だと判断した情報はすぐに音声化を止める。
2. 聞き逃した情報を聞き返せる。
3. キー操作に応じて速やかに読み上げる。
4. 速く正しく分かりやすく読み上げる。

また文字の読み上げに必要な機能として以下の機能をあげている[4]。

1. アルファベットの大文字と小文字は音声のピッチで区別する。
2. 漢字の読み上げには3種類のモードが必要である。かな漢字変換時は、漢字ごとに「タンボの田」などのユニークな説明を付加する。カーソル移動に伴って一文字ずつ読み上げるときは音読みまたは訓読みなどのような簡潔な読みを用いる。その他の場合は普通に読む。

斎藤が指摘しているように、同音異義語が多い漢字を音声だけで識別するためには、「タンボの田」のような「詳細読み」機能が必要であり、既存の音声化システムは「詳細読み辞書」を備えている。また日本語では英語の大文字と小文字の区別に加えて、ひらがなとカタカナ、全角文字と半角文字を区別する工夫が必要となり、声のピッチ(声の高低)を変えることで識別しているケースが多い。

2.2 音声化システムの問題点

DOS は文字ベースの OS であった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。そのため、Windows はなじみにくいと感じたり、使いにくいと感じる人もいる。

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。X Window System などの GUI はテキストの世界の上に別のレイヤとして存在しているだけで、ユーザは好みに応じてテキストツールでも GUI ツールでも UNIX を利用できる。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。

つまり、視覚障害者にとって DOS は OS としての機能が低く新たに習得しても報われない、Windows の GUI は音声化システムに本質的に適していない、UNIX には音声化システムがほとんどないといった状態になっており、最新コンピュータの機能を活用できる使いやすい日本語音声化システムが待望されている。

また仕事や勉強の場で出会う情報は日本語と英語が混在している場合が多い。しかし既存の音声化システムは日本語用に設計されているため英語を正しく音声化することが苦手であるので、日米 2 ヶ国語が混在した情報を正確に又わかりやすく音声化できるシステムも必要とされている。

3. Emacspeak

画面表示用の GUI と、聴覚を使う音声化シ

ステムの相性が悪いという問題を解決するひとつの方法が、Raman が開発した Emacspeak¹である。テキストベースのアプリケーションであり Lisp 言語のプラットフォームでもある Emacs エディタがスピーチサーバを制御する音声化システムが、Emacspeak である。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で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。

1. AUI (Auditory User Interface) [5]: 視覚が使えるユーザ、使えないユーザ、或いは手足が不自由なユーザなどのさまざまな状況にあるユーザに適合(adapt)して情報を伝えるためには、ユーザインターフェースをシステム自身から切り離して、ユーザに最適化したユーザインターフェースを自由に選択できるようにしなければならない。このような考えに基づいて Raman は音声化システムにおける聴覚専用ユーザインターフェースの重要性を指摘し、AUI をもった音声化システムとして Emacspeak を開発した。Emacspeak における AUI の例としてカレンダーモードがある。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。しかしカレンダーを単純に読み上げるだけではこのような情報は得られない。そこで Emacspeak はカレンダーを表示するモードにおいては、必要な情報を付加した上で、“Monday, April 1st, 2001”と音声化する。
2. Voice フォント: GUI 画面に表示される

¹ Emacspeak は GNU オープンソースとして <http://emacspeak.sourceforge.net/> で公開されている。

文字には複数のフォントがあり、さらに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。音声においても、声の主を変えたり、ピッチやアクセントなどの声の個性を変えたり、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。Emacspeak は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。たとえば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いのと同様な使いやすさを音声ユーザに提供している。

Emacs 用の Web ブラウザである W3 は ACSS (Auditory Cascading Style Sheets²)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。

3. **Auditory** アイコン: 文字メッセージは再生に時間がかかるので、いくつかのイベントに対しては短いサウンドを割り当てて迅速にユーザにイベントを通知する。

4. Bilingual Emacspeak Platform

4.1 BEP の概略

Raman の 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。そこで著者らは、1999 年に晴眼者と視覚障害者の共同プロジェクトとして Bilingual Emacspeak Project を立ち上げた。このプロジェクトは、(1) 日米 2 ヶ国語のソフトウェアスピーチサーバを Windows 用と Linux 用の 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。BEP がバイリンガルなのは、仕事や高等教育の場では日本語と英語が混在した情報を扱うことが多く、このような場合に英語を正確に音声化できるシステムがまだ存在していないからである。Windows をサポートしているのは Windows ユーザが多いからであり、Linux もサポートしているのはオリジナルの Emacspeak が Unix のみで動作していたのと、GNU/Linux は Windows に代わる障害者用の高機能な OS になる可能性があるからである[6]。

4.2 BEP の構成

次ページの図 1 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、日米 2 ヶ国語の音声合成エンジンを操るスピーチサーバ部の 2 つの部分に分かれる。

スピーチサーバは、日本語は Shift_JIS で受け取る³。Windows 用のスピーチサーバは G. Bishop のソース⁴を元にしており、Microsoft Speech API 4 に準拠した日本語及びアメリカ英語の音声合成エンジンを使用する。Linux 用のスピーチサーバはまだ日本語のエンジンしか使用できない。この日本語エンジンは、商用の音声合成 SDK⁵を基にして自主開発した。Linux には標準の音声合成インターフェースがないために日米 2 ヶ国語のエンジンを同時に制御することが難しく、Linux 版は英語はカタカナ英語、つまりカタカナで表記された発音で日本語エンジンが読み上げる。

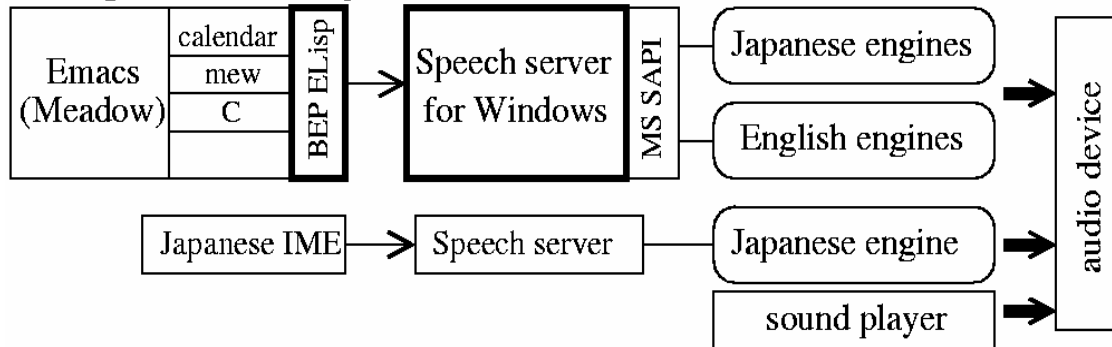
² Cascading Style Sheets の詳細は、<http://www.w3.org/TR/REC-CSS2> 参照。

³ 多言語の取り扱いを容易にするために、将来は Unicode または Emacs 内部の文字表現に変更する予定。

⁴ G. Bishop が私的な目的のために書いたソースをオープンソースとして譲り受けた。

⁵ クリエイトシステム開発株式会社の Linux 版日本語音声合成開発キット Ver1.0。

Bilingual Emacspeak for Windows (Voice Meadow)



Bilingual Emacspeak for Linux

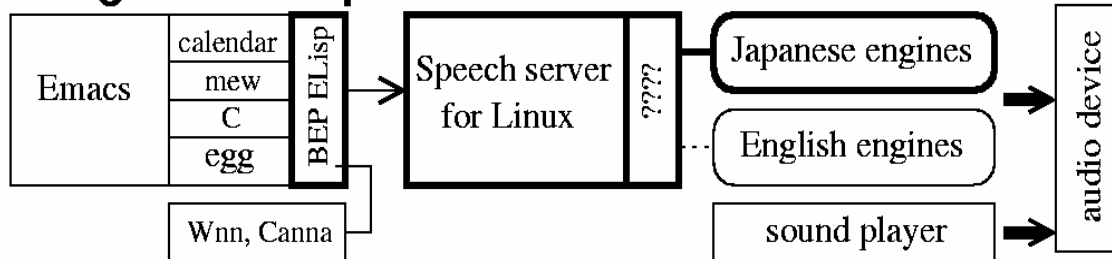


図 1. BEP の構成。太線枠が自主開発部。上半分が Windows 版、下半分が Linux 版の構成。Emacs Lisp (ELisp)部は両 OS で共通しているが、スピーチサーバは各 OS ごとに開発する。かな漢字変換の音声化は、Windows 版では Windows 用のスクリーンリーダに任せているのに対し、Linux 版では egg が担うかな漢字変換を BEP 内部で音声化する。図中、Meadow は Windows 用の Emacs 20.x、Mew はメールソフト(MUA)、Wnn 及び Canna はかな漢字変換サーバの名前。

Windows 版のスピーチサーバは、英語、日本語とも以下の 9 種類の声(Voice フォント)を持つ⁶。通常は(a)の声を使い、コメント文を読み上げるときは(b)の声を使うなどのように、情報の種類によって声を使い分ける。

- (a) 男性 1;
- (b) 抑揚のない単調な男性 1;
- (c) ピッチが高い男性 1;
- (d) 男性 2;
- (e) 男性 3;
- (f) 女性 1;
- (g) 女性 2;
- (h) 中性 1;
- (i) 中性 2;

現バージョンの BEP はスピーチサーバ内部で言語を自動判定しており、マルチバイト文字を含んでいる文字列は日本語の音声合成エンジンで読み上げる。このとき、同じ性格の声をもった日本語の声に切り替える⁷。

4.3 スピーチサーバのコマンド

スピーチサーバを制御するコマンドは、音声合成装置 DECTalk を制御する Emacspeak に準じている。主なコマンドを以下にまとめる⁸。

- **queue**; このコマンドに続く文字又は文字

⁶ Linux 版は日本語のみ。

⁷ たとえば英語の(b)の声で読み上げるべきときは、日本語でも(b)の声を使う。

⁸ Linux 版では実装中の機能もある。

列をキューに追加。

- **start queue**; キューを実行し順番に再生。
- **say letter**; このコマンドに続く一文字を読み上げる。
- **say word**; このコマンドに続く文字列を非同期で読み上げる。
- **stop**; 即時停止。
- **set voice**; 声の種類を指定。
- **set speech rate**; 再生速度を wpm の単位で指定。
- **tone**; 指定周波数と指定長さのトーンを作ってキューに追加。
- **play sound**; 指定サウンドファイルを再生。

4.4 英語音声化に対する BEP の工夫

既存の日本の音声化システムは、カタカナ英語以外に英語音声化に対する工夫をしていない。また、既存の日本語エンジンは英語を正しく発音する機能を持っていないので、発音が間違っているために何が書いてあるのかわからなくなるケースが生じる。ローマ字に関しても同様で、日本語エンジンはローマ字を読むのが苦手である。

BEP の Windows 版スピーチサーバは英語と日本語の音声合成エンジンを持っているので、英語は英語のエンジンでネイティブの発音で正しく読み上げる。Linux 版のスピーチサーバはまだ日本語のエンジンしか制御できないので、カタカナ英語の辞書⁹を使って、英語をカタカナに置き換えてから日本語のエンジンで読み上げている。現在、英語やローマ字をユーザにわかりやすいように音声化するための改良を行っており、詳細は 5.3 章で議論する。

4.5 BEP の現状

BEP は全盲の視覚障害者用の高機能な音声化システムを目指して開発中であるが、評価用バージョンを 2001 年 1 月に Web

⁹ この辞書は石川准教授が開発した音声化システム「グラスルーツ」の辞書を使っている。この辞書には、たとえば“English”という見出しに対し、「イングリッシュ」というカタカナで表記した発音を書いてある。

(<http://www.argv.org/bep/>)で公開し、メーリ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[7]。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ので現在のユーザ数は 10 人弱しかないが、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。現在、正式な一般公開を目指して、安定で完全なバイリンガル読み上げを行えるように改良中である。また Windows 用のスクリーンリーダーは高価なので、BEP のようなオープンソースの音声化システムの価値は高い。

5. 実用的な音声化システムへの要求

BEP の開発とユーザテストによって、視覚を使用せずに音声出力だけでバイリンガルな情報を得る際に重要な要求が、以下のようになり明らになった。

5.1 概論

視覚なしで聴覚だけに出力する際は、可能な限り迅速にそして正確に、必要な情報を音声化することが重要である。また人間が話すように自然に読み上げることができるシステムよりも、早口でも聞き取りやすいシステムのほうが大事である。つまり、(1) 正確な音声化、(2) 軽快で良好な応答性、(3) 高速でも聞きやすい声、(4) 聴覚に最適化したユーザインターフェース、(5) 本質的な情報だけを音声化、(6) ユーザの要求にこたえられる高機能とカスタマイズ可能性、といった条件を満たした音声化システムが必要である。

5.2 音声合成エンジンへの要求

1. 音声化開始や停止などの操作に対して応答性がよいこと。
2. 早口で読み上げても文字を識別しやすいこと：視覚障害者は、日本語エンジンの再生能力の上限である倍速で使用している。倍速で音声化すれば読み上げに必要な時間は半分になる。英語の場合

合、3 倍速以上で使用しているユーザもいるので、日本語においても 3 倍速で再生しても文字を識別しやすくして耳が疲れない音声合成エンジンが必要である。

3. 主要言語のサポート: 日本語、英語、西欧の諸言語、中国語、韓国語など、さまざまな言語のエンジンが必要である。
4. 統一インターフェース: Windows には Speech API という統一インターフェースがあり、日本語の特徴を考慮していないという欠点はあるけれども日本語のエンジンと英語のエンジン両者を簡単に制御できる。しかし Linux にはまだこのようなインターフェース(API)がないので、2 ヶ国語のエンジンを統一的に制御するスピーチサーバを書くのが Windows ほど簡単ではない。
5. 一文字読み上げ: 現在の音声合成のメカニズムは、一文字だけを明確に読み上げることが苦手である。しかし文章の内容を仔細に調べるときなどはカーソル下にある文字を一文字ずつ読み上げて聞き取る必要があるので、'い'、'ち'、'も'、'じ'ずつ明解に読み上げる機能が必要である。
6. 音素表現による読み上げ: この一文字読み上げのように、あらかじめ文字に対する発音が決まっている場合がある。このような場合は文字を音声合成エンジンに渡すより、音素で表現した発音をエンジンに渡したほうがエンジンの処理が軽くなるし、発音も完全に指定できる。
7. DECTalk のような高機能なエンジン: 多様な声と多様な音声修飾ができないと、さまざまな情報を区別して音声化するのが難しい。
8. (画面表示用にいちいちフォントを購入する必要がないように、)音声合成エンジンも簡単に(できれば無料で)入手できないと、音声利用が普及しない。
9. ローマ字や英語をなるべく正しく読み上げる機能も必要である。

5.3 日米 2 ヶ国語音声化に必要な機能

Windows 版のユーザテストの結果、英語の発音で読み上げると英語を聞き取れないユーザがいることがわかった。このようなユーザには、ネイティブ英語で読み上げるよりも、日本語エンジンでカタカナ読みしたほうがわかりやすい。ただし、ネイティブの発音で英語を正しく読み上げることが必要な場合もある。一般に日本語として情報を認知しているときはカタカナ英語で読み上げ、英語として認知しているときはネイティブ英語で読み上げる必要がある。目で読むときは頭の中でこれら 2 つのモードを自動的に切り替えることができるが、耳で聞くときは音声化システムのほうで切り替える必要がある。

この 2 つの英語発音モードは、コンテキスト及びユーザの要求に応じて切り替える必要がある。システムメッセージ、ディレクトリやファイル名、単語単位の英語などはカタカナ英語で読み上げ、文或いはパラグラフより長い単位の英語はネイティブ英語で読み上げる。

この要求を満たすために、Windows 版及び Linux 版両方のスピーチサーバを、カタカナ英語とネイティブ英語の両方を発音できるように改良中である。Emacs Lisp レベルでコンテキストに応じてどちらのモードを使うかを指示し、或いはユーザの指示に従って、2 つのモードを切り替えて音声化するようにする予定である。

英語読み上げ時に大事なもうひとつの点は、間違った発音で読み上げるくらいならスペルアウトしたほうがユーザに正確に情報を伝達できるということである。カタカナ英語モードの場合、カタカナ英語辞書で変換できなかった英単語はスペルアウトして読み上げる。ローマ字の場合は、どちらのモードでもカタカナに変換して読み上げる。

また、ネイティブ英語で読み上げる場合、英語の再生速度を日本語よりも低く設定できるようにする必要がある。

6. まとめ

Emacs が好きな晴眼者と視覚障害者のオープンな共同プロジェクトとして、バイリンガル音声化システム **BEP** を開発しており、オープンソースとして公開している。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及び **Raman** の **Emacspeak** を基にしており、日米 2 ヶ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。

全盲の視覚障害者によって、音声出力だけでもパソコンを利用できることが示されている。こうした利用を可能にするためには、(1) 正確、(2) 迅速な応答性、(3) 倍速以上の読み上げでも聞きやすいこと、(4) 聴覚に最適化したユーザインターフェースの使用、(5) 本質的な情報だけを音声化、(6) ユーザのニーズにこたえる機能の装備とカスタマイズ性、が必要である。

日本人用の日米 2 ヶ国語音声化システムには、2 種類の英語読みあげモードが必要である。目で見たとときに日本語として認知しているようなアルファベット文字列はカタカナ英語で音声化し、英語として認知している情報はネイティブ英語の発音で音声化する必要がある。また、間違った発音で読み上げるくらいならば、英語をスペルアウトして読み上げたほうがよい。

BEP はまだ開発中であるが評価用のバージョンを Web (<http://www.argv.org/bep/>) で公開し、完全なシステムとしての一般公開を目指してメーリングリストで活発な開発論議とサポートを行っている。

謝辞

誰よりもまず、井上浩一氏、坂本貢氏、高橋玲子氏、高橋直人氏など、ユーザ及び開発者として貢献し有益な助言を授けてくれた **BEP** のコアメンバー達に感謝する。また、オープンソースとして或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、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「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] 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 (1995).
- [3] 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).
- [4] 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 (1995).
- [5] Raman, T.V: Auditory user interfaces — toward the speaking computer —, Kluwer Academic Publishers (1997).
- [6] 渡辺隆行、井上浩一、坂本貢、切明政憲、他: Bilingual Emacspeak と視覚障害者の Linux アクセシビリティの向上, **Linux Conference 2000 Fall**, pp.246-255 (2000).
- [7] Watanabe, T., Inoue, K., Sakamoto, M., Kiriake, H., Honda, H., Nishimoto, T., & Kamae, T.: Bilingual Emacspeak Platform — A universal speech interface with GNU Emacs —, **Universal Access in HCI 2001**, Vol.3, pp.446-449 (2001).