

視覚障害者、盲ろう者向け音声・点字コンピュータ・ オペレーティングシステム GR for UNIX の開発

2005 年 4 月 25 日

静岡県立大学国際関係学部 教授 石川 准
有限会社サイパック 取締役社長 工藤 智行
学習院大学法学部政治学研究科 南谷 和範

あらまし: 音声・点字を基本的ユーザインターフェースとするコンピュータ・オペレーティングシステムを Linux プラットフォームを対象に開発しており、その第一歩として実用的な Linux スクリーンリーダを開発した。

キーワード: Linux、スクリーンリーダ、支援技術、アクセシビリティ

はじめに

本プロジェクトは視覚障害者、盲ろう者向けに、聴覚(音声合成)と触覚(点字ディスプレイ表示)をユーザインターフェースの基本とするコンピュータ・オペレーティングシステムを Linux を基にして実現しようとする試みである。プロジェクトの成果は視覚障害者、盲ろう者が日常生活や就労に際して日々利用可能なシステムとして使用されることを想定している。そのために、専門的な知識が無くても視覚障害者、盲ろう者が自力でインストールが可能な、日々の作業環境として実用的なディストリビューションパッケージを作成することが本プロジェクトの最終的な目標である。

本プロジェクトではコマンドを、聴覚・触覚のユーザインターフェースによって操作できるようにし、Linux を始めとする UNIX で動作する日本語対応のスクリーンリーダを開発した。本稿では、スクリーンリーダの開発経緯と、今後の開発の方向性について解説する。なお、本プロジェクトは厚生労働科学研究費補助金の助成を受けて開発を行っている。

目的

コンピュータは情報を画面に視覚的に提示し、人はポインティングデバイスで「選択」する。今日のコンピュータと人のインタラクションは大部分がその反復である。このインタラクションは GUI と呼ばれるインターフェースにより実現されている。ところが視覚障害者は音声または点字での情報提示を必要とする。だが自ら音声や点字表示をサポートするソフトウェアは極めてわずかである。そこで現実問題として VUI (Visual User Interface) を AUI (Auditory User Interface) および BUI (Braille User Interface) に変換するソフトウェアがどうしても必要になる。それが GUI デスクトップ用スクリーンリーダである。

かつての CUI (Character-based User Interface) AUI/BUI 変換は単純であったためあまり問題はなかった。だが、今日の GUI AUI/BUI 変換では視覚障害者に効率的な操作環境を提供することは難しい。そもそも視覚、聴覚、触覚はそれぞれ異なる特性を持つ感覚器であるから、本来それぞれの UI は独立に設計されるべきものだが、現実はそのようになっていない。視覚障害者は、スクリーンリーダというストローの穴から画面を見て選択せざるをえない。しかもこのストローは自由自在には動いてくれない。

現在、一般のコンピュータ環境においてはマイクロソフト株式会社の Windows を中心とする GUI デスクトップ環境が圧倒的なシェアを有している。これに伴いこの十年近くは、視覚障害者も GUI を音声や点字出力で表現するスクリーンリーダを用いて Windows 等の GUI を利用することを余儀なくされている。だが、現段階の GUI オペレーティングシステムは GUI を暗黙の前提として設計されており、他のユーザインターフェースを想定して作られてはいない。このためスクリーンリーダを用いる場合、GUI をいわばむりやり音声や点字に変換して表現することになり、こうしたコンピュータ環境は視覚障害者にとって、最良の環境とは言いがたい。

GUI のデスクトップ環境を音声合成と点字により情報提示するスクリーンリーダとしては、Windows 環境では JAWS¹ や PC-Talker² 等の商用製品が国内でも多く利用されている。Linux の環境では、GNOME 環境用のスクリーンリーダとして米国サン・マイクロシステムズ社が開発を行っている GNOPEPERNICUS³ というシステムがあるが、開発途上であり実用段階には至っていない。このようなスクリーンリーダはいずれも GUI を対象としたものであり、前述のような問題、つまりユーザビリティにおける限界を抱えている。

一方、視覚障害者の中には、未だに MS-DOS を MS-DOS 用のスクリーンリーダと共に利用しているユーザが少なからず存在する。また Windows 環境でも DOS ボックス内でコマンドプロンプトを中心に操作しているユーザも多い。彼らを単なる「GUI 不適合者」と見なすのではなく、現在の GUI デスクトップ環境には視覚障害者の利用障壁があると考えるのが我々の立場である。このような MS-DOS ユーザは、一方で今日的なアプリケーションが使用できないという大きな問題を抱えている。MS-DOS が主流の OS の座から退いて既に十数年が経過しているが、受け皿となる環境が見当たらないのが実情である。

ただし、高いコンピュータスキルがないと使いこなせないという限界はあるものの、Linux のコンソール環境でならば、既に欧米では BRLTTY や Emacspeak により視覚障害者にとって実用的なコンピュータ環境が実現されている。

また、点字ディスプレイと音声出力を搭載した WindowsCe や Linux ベースの PDA が製品化され多く利用されている。

これらのプロダクトの多くは独自のアプリケーションを搭載し、最初から AUI と BUI の特性をうまく利用しているため、視覚障害者から歓迎されている。

なお、日本国内では ASUKA という Linux のスクリーンリーダを実現しようとするプロジェクトが存在したが、

¹ http://www.freedomscientific.com/fs_products/JAWS_HQ.asp

² <http://www.pctalker.net/>

³ <http://www.baum.ro/gnopernicus.html>

開発成果は一般公開されておらず、開発もすでに打ち切られている。Emacspeakの多言語化を目指した先駆的なプロジェクトBEP(Bilingual Emacspeak Project)⁴の成果物はこの分野で一定のユーザを確保したが、Emacs環境に限定された音声合成のみのシステムである。

以上のような実情を鑑み、我々はLinuxを始めとするUNIXの非常に充実したCUIのコンソール環境を対象に音声合成や点字表示の機能を追加するべく作業を開始した。将来的には聴覚、触覚に基づくシェルや視覚障害者向けのエディタ、ブラウザ、メーラ等の統合環境を開発することで、実用に耐えうる視覚障害者、盲ろう者向け音声・点字コンピュータ・オペレーティングシステムを実現する。

我々は、プロジェクトの起点として、まずコンソール環境、コマンドライン環境を対象として日本語が扱え、音声合成や点字表示の両機能を備えたスクリーンリーダを開発することにした。当初は日本語のみへの対応だったが、英語を始めとする複数言語に対応できるような内部構造を実現することも、一つのテーマとした。

一般にアメリカの視覚障害者は音声合成を使ってコンピュータを利用しており、ヨーロッパの視覚障害者は点字ディスプレイを使ってコンピュータを利用する傾向にあるが、日本では音声ユーザが多いものの、点字ユーザも一定以上存在するという地域状況にある。そこで、単に点字出力と音声出力に対応させるだけでなく、どちらか一方のみでも十分な操作環境を提供することを課題とした。

視覚障害者の間でも点字出力と音声出力のいずれがユーザインターフェースとして望ましいかは意見の分かれる問いである。点字出力は概して複雑な情報を把握するのに有効であるが、反面、点字の習熟が必要である。また、点字出力を行う点字ディスプレイは高額な製品(1台50万円以上が相場)であり、誰もが手軽に入手できるものではない。一方、音声出力は大量の情報を短時間で把握する場合に大きなメリットを発揮するが、ユーザ主導で二次元的な画面情報を探索するような局面では不利となる。このように、ユーザインターフェースとしての点字出力と音声出力は一長一短の関係にある。

スクリーンリーダを日本語対応にするには複数の要素を考慮する必要がある。日本語の文字コードへの対応はもちろんであるが、それ以外に、日本語自動点訳エンジンへの対応、日本語音声合成エンジンへの対応、日本語インプットメソッドへの対応等が必要となる。また、日本国内で多く使われている点字ディスプレイへの対応も必須である。

幸い、日本語の点字を扱うために必須の点訳エンジンは、石川の開発したExtra for UNIX⁵が存在する。これはEUC-JPの日本語テキストを入力として与えると、点字形式の出力が得られるエンジンである。これを本プロジェクトの点訳エンジンとして使用することにした。

一方、日本語の音声合成システムはWindowsや組み込みシステム向けのものはいくつか存在するが、Linux向けのものは少数しか存在しない。

⁴ <http://www.argv.org/bep/>

<http://lc.linux.or.jp/lc2002/papers/watanabe0920h.pdf>

⁵ http://www.extrasoft.jp/product_2.html#A002

なお、多くの日本語インプットメソッドではユーザが変換したい文字や単語の読みを入力し、候補の中から適切なものを選択する。ところが音声で文字や単語を読み上げた場合、同音異義語の区別ができない。同音異義語の区別問題は仮名分かち書きで表現される日本語点字にも当てはまる。この問題に対応するために「詳細読み」という方法で変換候補として表示する文字がどのような漢字であるかを解説する。たとえば「日本語」という単語であれば「日曜日の『日』、本の『本』、語るの『語』」のように読み上げ、点字出力を行う。

また、日本国内ではケージーエス株式会社の点字ディスプレイが大きなシェアを持っている。国内での使用を前提する場合、ケージーエス株式会社の点字ディスプレイのサポートは必須といって良い状況である。

なお、GR for UNIX という命名には、かつて石川が MS-DOS 用のスクリーンリーダーとして開発したグラスルーツ(通称 GR)を UNIX 上で実現するという意味がある。

研究方法

我々は平成 15 年度の成果を基に、既存の欧米版スクリーンリーダーの移植や改良といったアプローチの限界を確認し、本格的な Linux 用スクリーンリーダーをスクラッチから開発するというアプローチを採用した。

平成 15 年度の成果

Linux 用のスクリーンリーダーの開発にあたって BRLTTY の存在に注目した。BRLTTY は欧米言語を点字ディスプレイで出力することを主眼に開発されてきたスクリーンリーダーである。すでに十年近い開発実績があり、欧米の視覚障害者を有する Linux ユーザの間では広く知られたソフトウェアである。国内でも視覚障害者が Linux を利用する場合に標準的に利用されている。ただし、欧米言語の点字出力にフォーカスした設計であり、日本語への対応は考慮されていない。また、画面内容の音声による出力はあくまで補助的な機能としてサポートされている。

我々は BRLTTY Version 3.41 を出発点に独自の拡張を加えて日本語スクリーンリーダーを試作することをテーマとし、平成 15 年度の開発を行った。試作したスクリーンリーダー部分を BRLTTY Plus と名付け、石川のホームページおよび sourceforge.jp⁶にて GR for UNIX というプロジェクトとしてソースコードと関連情報を公開している。

オリジナルの BRLTTY は、Linux の仮想コンソールデバイスである `/dev/vcsa` から画面内容を読み出し、点字出力を行うソフトウェアである。ところが、`/dev/vcsa` は日本語文字コードを正しく扱える設計になって

⁶ <http://grunix.sourceforge.jp>

おらず、また、この方法では日本語インプットメソッドの詳細読みを実現するのが困難であった。幸い、BRLTTY には、`/dev/vcva` を使用せず、仮想端末ソフトウェアである GNU Screen 内のイメージを共有メモリを使って読み出す対象とする試験的なパッチが存在した。我々はこのパッチを開発の出発点として利用することにした。

オリジナルの BRLTTY は、点字ディスプレイのボタン以外からのコマンドを受け付けられない構造になっており、点字ディスプレイを接続していない PC では起動さえできないという問題があった。音声合成を主として利用するユーザにとっては、キーボードからスクリーンリーダに対する指示が送れないことになる。そこで、BRLTTY Plus ではコマンドをプロセスの外部から送れるように IPC を用いたインターフェースを新たに設けることにし、併せて点字ディスプレイが接続されていなくても BRLTTY が動作するように内部構造を変更した。また、Screen からはキー操作によって BRLTTY Plus に対してコマンドを送れるような拡張を行った。

さらに、日本語の点字を扱うために前述の EXTRA for Unix 自動点訳エンジンを組み込み、日本語の音声合成を行うために、クリエートシステム開発株式会社の Linux 音声合成ライブラリ用のドライバを開発した。また、ケージーエス株式会社のブレイルノート 46X 用のドライバを開発し、従来より BRLTTY がサポートしていた ALVA サテライト、パワーブレイル等の点字ディスプレイと合わせると、日本国内で使用されているほとんどの点字ディスプレイをサポートすることができた。

日本語かな漢字変換入力時の音声による詳細読みの提示機能を実現するために、詳細読みのライブラリも開発した。また、`canfep`⁷ という Canna のクライアントに詳細読みライブラリと、BRLTTY Plus と通信するための IPC インターフェースを組み込むことで、同音異義語を詳細読みで情報提示しながら漢字入力を行えるようにした。

BRLTTY Plus 環境の起動手順は次のとおりである。

1. 音声・点字出力を行うデーモンとして BRLTTY Plus 本体を起動する。
2. コンソール上で日本語入力を行うための `canfep` を立ち上げる。
3. さらに共有メモリにより画面情報を提示し、BRLTTY Plus と通信できるように変更した GNU Screen を起動する。

実際には、BRLTTY Plus のデーモンはシステムの起動スクリプトで起動し、`canfep` と Screen はユーザのログオンスクリプトから自動的に起動されるようにした。

平成 16 年度の成果

平成 16 年度は日本語 Linux スクリーンリーダを実用的なレベルに引き上げることを目標とし、前年度の

⁷ <http://www.geocities.co.jp/SiliconValley-Bay/7584/canfep/>

成果を元に本格的なスクリーンリーダの開発を目指すことにした。最低限必要なスクリーンリーダ機能は実現済みであるが、カーソル位置の文字の読み上げ機能や、削除文字の読み上げを始めとする、多くのきめ細かな機能の実現、音声合成機能の充実等は課題として残されていた。

平成 15 年度は、BRLTTY を元に開発を行ったが、BRLTTY には我々が目標を実現する上で、いくつかの構造上の大きな問題点が存在することが明らかとなった。

欧米言語では US437 やこれに類する点字体系を採用することで、画面に表示される英数字や記号類と点字ディスプレイに出力される点字に 1 対 1 の関係が維持できる。たとえば、1 行 80 桁の標準的な VGA コンソール環境であれば、80 桁の点字ディスプレイに対しては単純なテーブルを用いて 1 文字ごとに変換し、1 行分の内容全部を出力できることが保障される。しかし、仮名分かち書きを行う日本語点字の場合、1 対 1 の関係は保障されない。たとえば、「南」という文字は、文脈によって「なん」(2 桁)あるいは「みなみ」(3 桁)という点字出力を行う場合があり、必要な桁数が変動する。BRLTTY はこのような桁数の変動を考慮していない。

前述のように BRLTTY は音声合成機能を実現しているものの、音声合成に対する内部構造が貧弱であり本格的な用途には耐えない。各点字デバイス、音声合成デバイスに対する処理が直列的に実現されているために、音声合成中は点字ディスプレイからの操作がまったく行えない等の致命的ともいえる問題が明らかとなった。BRLTTY Plus では読み上げの中断機能という音声合成のスクリーンリーダにとって不可欠な機能を実現したが、BRLTTY の内部には読み上げの中断を行うための構造が存在しないため、IPC のインターフェースを音声合成ドライバが直接覗く、極めて不自然な形で実現せざるを得なかった。

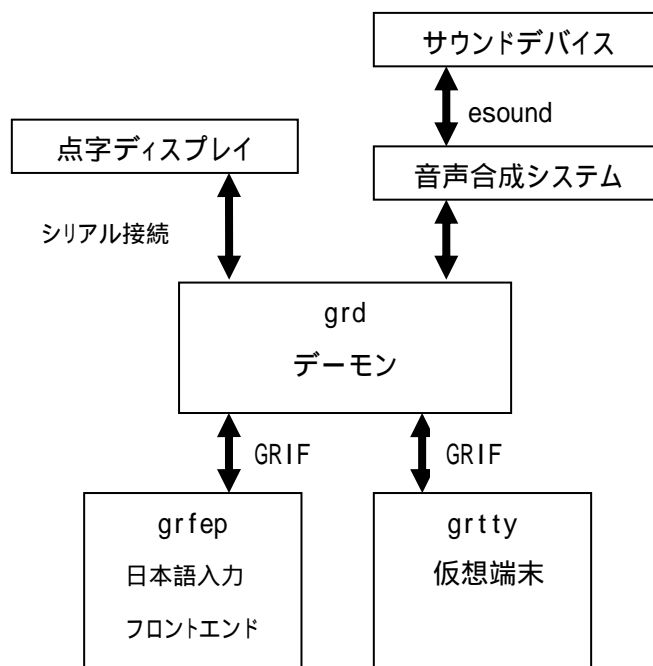
我々は、外部のアプリケーションとスクリーンリーダが密に連絡を取りながら音声合成や点字表示を実現するモデルを描いているが、BRLTTY はこのようなモデルが実現できる基本設計となっていない。BRLTTY を元に引き続き開発を進めることは不可能ではないが、音声合成機能の充実や、本格的な日本語対応を行うには、基本構造を含む内部構造の大幅な手直しが必須であることが明らかであった。

また、BRLTTY のクライアントとして使用した Screen はデフォルトで Control+A をタイプした後に、コマンドキーを入力しなければならないという制約がある。MS-DOS や Windows のスクリーンリーダは、1 キーの入力でスクリーンリーダに対するコマンドを送ることができるので、これらと比較すると明らかに操作性の面で劣っている。また、Screen を使用する限りにおいては、削除文字を読み上げる機能の実現が極めて困難であることも明らかとなった。

以上のような観点から BRLTTY と Screen を元にして日本語対応の本格的なスクリーンリーダを開発するのは困難と判断し、BRLTTY の問題点を見据えた上で、BRLTTY Plus の経験を生かし、まったく新たな設計に基づいてスクラッチからの開発を行うこととした。設計に際しては、将来的な国際対応を視野に入れ、日本語固有の構造を避け、多言語環境が実現できる構造とした。また、Linux 固有の実装はできるだけ避け、他の UNIX システムでも動作できるような配慮を行った。実際に、Mac OS X でも使える音声合成エンジンの種類は異なるものの、ほぼフル機能が実現できている。

今回開発したスクリーンリーダは grd, grfep, grtty の三つのソフトウェアから構成される。grd はスクリーンリーダの中核を担うデーモンであり、grfep が日本語入力用のクライアント、grtty が仮想端末という役割分担となっている。BRLTTY Plus における BRLTTY デーモンが grd で、Screen が grtty に相当する。開発したそれぞれのコンポーネントの詳細は次のとおりである。

スクリーンリーダ構成図



grd

GR for UNIX のシステムの中核を担うデーモンである。GRIF というインターフェースを通じて音声読み上げおよび点字表示サーバ機能を提供する。GRIF 対応クライアントからのリクエストに応じ、音声読み上げや点字表示、コマンドの実行等を行う。クライアントとの GRIF インターフェースは、UNIX ドメインのソケットを使用し、各クライアントとの通信専用スレッドを作成することで対応した。さらには、後述の grtty 等の仮想コンソールソフトからは、共有メモリを通じて画面情報を受け取る。

また、チャンネルと呼ぶ複数の仮想画面を扱えるようにし、スクリーンリーダの読み上げ対象画面をいくらかでも拡張でき、音声読み上げや点字表示は、複数の「チャンネル」から一つを選んで行える構造とした。ヘルプ機能はこのチャンネルを使用して実現したが、将来的にクリップボード用のチャンネルや、コマンドメニュー等への応用も可能である。

grd は点字ディスプレイからのコマンドの監視、音声合成、GRIF サーバの三つの役割をそれぞれ単独のスレッドを作成し実行する。このため特定のデバイスの処理により、別のデバイスが待たされることがない。複数の音声合成ドライバや、点字ディスプレイデバイスを同時に利用できる構造になっている。

grd では複数の音声合成ドライバをサポートした。日本語音声 TTS エンジンとして、クリエートシステム開発株式会社の Linux 音声合成ライブラリ用のドライバを開発した。また、英語 TTS エンジン用のドライバとして、Festival と AT&T Natural Voice 用のドライバを開発した。さらに、Mac OS X で grd を動作させるために、日本語と英語をサポートしているクリエートシステム開発株式会社の Document Talker 用のドライバも併せて開発した。

音声出力は生成された PCM データを esound(ESD)を経由して、デバイスに出力することで音声デバイスの占有を防ぐ構造としたが、esound の動作が著しく不安定であることが、音声ドライバ開発を非常に難しくした。esound の代わりとなる、ミキサー機能の選定が今後の課題といえる。

点字ディスプレイを使用するためには、各社の点字ディスプレイに対応した点字ディスプレイドライバが必要になる。grd では、石川の開発した視覚障害者用の Windows 上のエディタ、ブラウザ、メーラである Altair for Windows の点字ディスプレイドライバ部分をライブラリとして独立した構造に切り離し、UNIX に移植することで対応した。

grd では EXTRA for Unix 自動点訳エンジンを組み込んで点訳を行っているが、日本語のみに依存した構造を避けるため、他の点訳エンジンを組み込むことが可能な構造とした。

grtty

grtty は今回独自に開発した仮想端末ソフトウェアである。grtty は、疑似ターミナルと、vt100 エミュレータ、GRIF クライアントから構成される。grtty 経由で起動したシェルの操作内容は、grd に伝えられ、音声読み上げや点字表示が行われる。また、現在、画面に表示されている内容や、過去に表示された内容を、grd 側のレビュー機能により音声読み上げや点字表示することができる。

grtty は別端末で複数起動が可能なので、読み上げ対象とする仮想端末を切り替えながら利用することができる。コンソールで ALT+ファンクションキーによる複数の tty の切り替えを行えば、GNU Screen による、複数仮想端末の起動と同等の効果を得ることができる。

ファンクションキーや、PgUp/PgDn 等の特殊キーを、スクリーンリーダーのレビュー機能に割り当てることができた。Ctrl+A に引き続く 2 ストロークのキー入力を必要とする、Screen を使用した場合と比べて操作性が格段に向上した。

とはいうものの、ファンクションキーや特殊キーをレビュー用のキーに割り当てると現在のスタイルではユーザビリティに限界がある。一般のスクリーンリーダーではテンキーをレビューに用いるのが常道となっており、現在のキーアサインはユーザビリティとしてはまだ不十分と感ぜられる。この部分の改善が今後の検討課題である。

grfep

ユーザが grfep を使って入力したアルファベット文字列は、Canna のサービスにより漢字列に変換される。

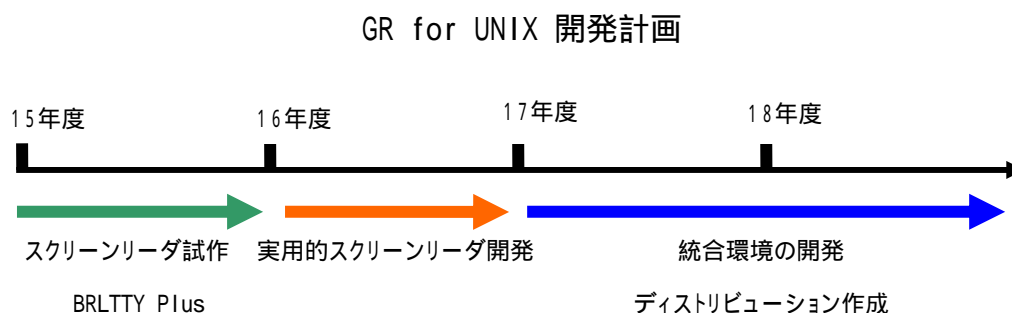
grfep は候補漢字列を GRIF 経由で grd に送る。grd は与えられた漢字列の詳細読み情報の音声読み上げと点字表示を行う。あらかじめ、grfep を起動した上で grtty を exec により起動すると、grtty の環境下での日本語入力が可能になる。Canna 以外の日本語インプットメソッドへの対応については今後の課題である。

以上のような開発を行うことで、前年度に実現した Linux スクリーンリーダを上回る機能と操作性を備えたスクリーンリーダが実現できた。ドキュメント等の整備が済み次第 grd, grtty, grfep はフリーソフトウェア GR for UNIX Version 2 として商用コンポーネントを除く部分を公開する予定である。

結論

我々のプロジェクトにおいて、スクリーンリーダの開発は最優先で実現しなければならないタスクであった。これが実現できた意義は大きい。このスクリーンリーダに搭載した API を用いて、エディタ、インターネットブラウザ、メールソフトウェア、ファイルマネージャ、そして配布パッケージの開発を順次行うことで、視覚障害者にとって、操作しやすく、しかも効率的なコンピュータ操作環境を実現できる手応えを得た。

今後の計画



本プロジェクトは、専門的な知識が無くても、視覚障害者、盲ろう者が自力でインストールが可能で、日々の作業環境として実用的なディストリビューションパッケージを作成することが最終的な目標である。

今後は(1)スクリーンリーダの機能拡充 (2)視覚障害者の利用に特化したエディタ、ブラウザ、メーラ等統合環境の整備 (3)音声、点字対応の独自 Linux ディストリビューションの作成が大きなテーマである。

スクリーンリーダの機能拡充としては、ユーザビリティの向上を中心とする機能追加の他、esound を置き換える音声ミキサーソフトウェアへの対応、さらなる音声合成エンジンへの対応等が課題である。また、音声・点字出力のためのミドルウェアとして、他のアプリケーションにサービスを提供する機能である GRIF が実装済であるが、実際にさまざまなアプリケーションに組み込むことで GRIF を充実させることも一つの課題

である。現在 GRIF は UNIX ドメインソケットと共有メモリ、セマフォによるインターフェースであるが、将来的には TCP/IP での接続も可能にしたい。このためには、共有メモリ上で扱っている情報を TCP/IP のソケットを通じて差分を送るようなプロトコルの拡充が必要である。現在の grd は X86 のバイナリで市販されているライブラリをリンクしているが、TCP/IP での接続をサポートすることで、他の CPU アーキテクチャのプラットフォームへの対応に道が開ける。

Linux のコマンドラインインターフェースは機能が豊富であるが、利用するにはそれなりの知識が要求される。これは一般の利用者にとっては障壁の一つであるので、聴覚、触覚シェルの開発が不可欠である。また、視覚障害者の利用に特化したエディタ、ブラウザ、メーラ等統合環境を整備することで、日常的に就労や研究等に利用できる環境とすることができる。ウェブブラウジング、文書編集、メールの読み書きが現代における PC の利用形態の大半を占めるので、これらを視覚障害者の利用に特化したユーザインターフェースで提供できればメリットは非常に大きい。また、この分野においては、JavaScript や XML 等にも対応した音声・点字インターネットブラウザの機能強化が大きな課題である。

なお、我々は GUI スクリーンリーダや視覚障害者用 GUI デスクトップ環境を開発する意思はなく、代わりに CUI 用スクリーンリーダと視覚障害者の利用に特化したアプリケーションの組み合わせにより、一般のコンピュータユーザにとっての GUI デスクトップ環境に匹敵する環境を実現する。

以上のような開発内容を含む音声、点字対応の独自 Linux ディストリビューションの作成と配布が、プロジェクトの大きな目標である。このディストリビューションは、障害者が独力でインストールできるようなシステムにしたいと考えている。近年注目を浴びている 10D Linux 等の動向にも十分注目して、実用的な Linux ディストリビューションを開発したい。

さらに、システム全体の有効性、効率性に関する評価研究も並行して行っていく予定である。実際に今年度からは、GR for UNIX のユーザグループを設け、ユーザからのフィードバックを通じてシステムの改善を行う予定である。

参考文献

[1] 石川 准、工藤 智行 「Linux 日本語スクリーンリーダの開発」(平成 16 年 10 月 28 日 福祉情報工学研究会主催 第 22 回研究会音声研究会・福祉情報工学研究会にて発表)

[2] 石川 准 “Developing an AUI and BUI-Oriented Computer Operating System” (平成 17 年 3 月 18 日 カリフォルニア州立大学ノースリッジ校障害者センター主催 第 20 回テクノロジーと障害者国際会議にて発表)