

## 文書読み上げシステムの構想

4R-10

前川 拓也 小野山 隆  
日立ソフトウェアエンジニアリング（株）

### 1. はじめに

近年のコンピュータの普及により、高齢者や身体障害者による利用も増加した。しかし、このような人達が快適に利用できるようになるにはまだ多くの障害がある。特に GUI 主体のコンピュータ環境は、視覚障害者には扱いにくい。視覚障害者用の音声出力システムは多く出ているが、それらは盲人向けの音声主体のものが多く、視力の弱い人達にはかえって扱いにくい場合がある。

そこで、音声出力と画面表示とを組み合わせたシステムについて検討を行った。

### 2. 現状での問題点

読み上げを行うシステムはこれまでも様々なものが出ているが、これらの調査を行ったところ、以下のような問題点があった。

第一に、読み上げ位置のスキップができない、またはスキップの操作生が悪い点である。読み上げを聴き漏らしたときに読み上げ位置を戻す機能は必要である。また、改行コードごとのスキップ機能を持ったシステムもあったが、改行コードは文の途中にも入っているため、そこからの読み上げでは意味がとりにくい場合が多い。

第二に、現在読み上げている位置を知ることができない点である。音声だけではどこを読み上げているかを知るのは難しい。特に読み上げスキップを行うと、文書の流れが把握できなくなってしまう。

そこで、以上の問題点を解消する機能を考案し、その実現方法について検討した。

### 3. システムの特徴

#### (1) 読み上げ位置スキップ

読み上げのスキップ機能は、文書のある部分を繰り返し読み上げたり文書の内容を理解するうえで重要な機能である。実際にスキップを行う際には、その移動量がスキップの操作生に関わってくる。構文を無視した移動を行うと、文章の途中から読み上げを再開したとき、その文章の意味をとりにくい。また、大きな単位でのスキップ機能も必要である。

そこで今回のシステム試作では、句読点単位と画面1ページ単位のスキップを取り入れた。

#### (2) 読み上げ位置の表示

読み上げの際、特に読み上げ位置のスキップを行う場合には、読み上げ位置を確認できると非常に便利である。視力が0でない人はある程度画面も見ているので、画面上に文書を表示し、読み上げた部分にマークをつけていくようにした。

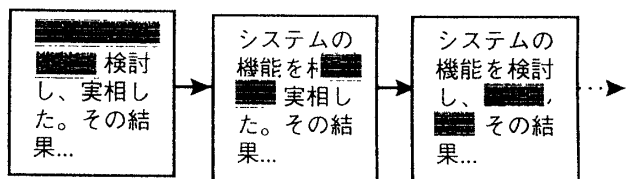


図1 画面表示

このように、読み上げた部分に随時マーク（反転表示など）をつけていくことで、読み上げ位置やスキップ時の移動量を視覚的に確認でき、システムの操作性を向上させることができる。

### 4. システム構成

本システムでは文書のある規則にしたがって切り分け、文字列リストを生成する。この文字列リス

The plan of text reading system

Takuya Maekawa, Takashi Onoyama

Hitachi Software Engineering Co.,Ltd.

トをデータ管理部に渡して読み上げ処理を実行する。

データ管理部では文字列リスト中の現在の読み上げ位置を管理し、指示に従って読み上げ位置を移動する。通常は文字列リストの要素ごとに読み上げ位置を順に進める。

またここでは文字列の画面上での位置も管理し、文字列の位置が画面の最下行である場合は、文書出力処理を呼び出して新しいページを表示する。そして、音声出力処理で文字列を音声合成装置に送信し、読み上げが終了次第マーキング処理を呼び出す。ここでは画面表示されている文字列にマークをつけていく。

読み上げ処理中にユーザからの入力があると、キー入力解析処理を行う。ここでユーザからの指示が読み上げ位置のスキップであれば、データ管理部に指示を送りスキップを実行する。

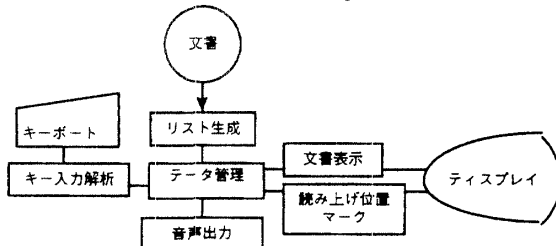


図2 システム構成図

## 5. 文書の管理方法

漢字は1文字だけではその読み方を特定できないことが多く、また漢字は2バイト以上のコードからなり、1文字づつ戻るような処理は簡単ではないので、1文字単位の管理はあまりメリットがない。そのため、読み上げ開始前に文書のある程度の長さで区切ってリストを生成することで、読み上げ位置の移動をリストの要素ごとに行えるようにした。

一方、現在の読み上げ位置をより詳細に画面に反映させるには、管理する文字列の単位は短い方がよい。また、文字列の切れ目を意味的な切れ目にあわせておかないと、誤った読みを行う可能性がある。

これらのことから、文節単位の文書管理が最適で

あると考えた。

しかし、文節単位の管理は高度な構文解析を必要とするので、マシンの負荷も大きくなる。

そのため、今回の試作では以下のような簡単な規則によって文書を切り分けることにした。

- ・句読点によって区切る
- ・数字はその塊で切り出す
- ・アルファベットは単語で切り出す
- ・' 'で区切る
- ・記号はその塊で切り出す
- ・改行コードでは区切らない
- ・空行によって区切る

この程度の規則であれば、処理の負荷もそれほど大きくはならず、ほぼ意味的な切れ目で文書を区切ることができる。

## 6. おわりに

読み上げシステムに今回提案した機能を実装することで、より高い操作性が期待できる。

しかし、現在の文書管理方法では切り出した文字列が長すぎたり、単位も含めた数字の読み上げ方など、多くの問題がある。また、段落や章ごとのスキップも実装したいが、そのためにはより高度な解析が必要となり、構文情報をデータとして保持する必要があるだろう。これらは今後の課題としてさらに検討する必要がある。

## 参考文献

- [1] 齊藤 正夫:「視覚障害者支援ソフトウェアの製作」 情報処理 (1995.12)
- [2] 斎藤 裕美, 野上 宏康:「日本語ワードプロセッサにおける自然言語処理」 情報処理 (1993.10)
- [3] 田中 穂積:「自然言語解析の基礎」 産業図書 (1989)