

Webベースコースウェアのための音声入力システムの開発

西村 竜一[†] 梶田 将司^{††} 武田 一哉^{†††}
板倉 文忠^{††} 鹿野 清宏[†]

本論文では、Webベースのオンライン教育環境に音声入力機能を加える WebSPEAC (Web SPEech Acquisition for Courseware) System を提案する。これまで、Webベースオンライン教育環境では、音声を出力する教材の作成は可能だったが、音声入力のできる教材やコミュニケーションツールの提供はできなかった。本システムは、(1) Webブラウザには音声入力機能のみを提供し、音声を用いるアプリケーションプログラムは Webサーバ上で一括管理するので保守性が高い、(2) クライアント側は、簡単な初期設定のみで利用できる、(3) サーバプッシュを利用して Webでの音声入力インタフェースを実現している、という特徴があり、音声を用いたインタラクティブな Webベースのオンライン教材の作成を可能にする。また、本システムの応用例として、Webベースの入力音声分析ソフトウェア、音声認識システム、受験者認証システムを作成した。このうち、音声入力部分について、入力音声分析ソフトウェアを利用してファイルアップロードによる従来システムとの比較実験を行った。その結果、本システムはステップ数、誤りステップ数、タスク完了時間において操作コストを削減できることが確認できた。

Development of Speech Input System for Web-based Courseware

RYUICHI NISHIMURA,[†] SHOJI KAJITA,^{††} KAZUYA TAKEDA,^{†††}
FUMITADA ITAKURA^{††} and KIYOHICO SHIKANO[†]

This paper proposes WebSPEAC (Web SPEech Acquisition for Courseware) System that provides a speech-input function for Web-based online education system. WebSPEAC has the following three features; (1) easy to maintain the system by the use of a simple speech input program in Web browsers, while the application program with speech processing is performed on the Web server, (2) a simple installation on client machines, (3) a speech-input function for Web-based system by using the server push technology. To show the effectiveness, a Web-based speech analysis system, a speech recognition system and a student identification system are developed as the examples of WebSPEAC System. By using the part of speech input method of the speech analysis system, it is clarified that WebSPEAC System can significantly reduce task steps, error steps and elapsed time for speech-input task, compared with the conventional method (file upload).

1. ま え が き

近年の計算機やネットワークの発達により、教育用オンラインシステムの開発が以前にも増してさかんになり、高度化する傾向がある。特にインターネットの

普及は、家庭やモバイル環境でも安価で高速なネットワークを構築することを可能にした。このことは、インターネットをベースとした遠隔教育システムの提供が容易にできることを示している。

教育用オンラインシステムの代表的なメリットがマルチメディア教材の利用と遠隔教育である。画像や音、シミュレーションなどのコンテンツを取り込んだマルチメディア教材は、これまでのテキストをベースとした紙教材にとってかわる新しい教材になる可能性がある。また、遠隔教育は、いつでもどこでもオンデマンドに学習でき、高い学習効果を期待できる。

このような状況の中、名古屋大学情報メディア教育センターでは、カナダのブリティッシュコロンビア大学で開発されたマルチメディアコースウェア作成支援環

[†] 奈良先端科学技術大学院大学情報科学研究科
Graduate School of Information Science, Nara Institute of Science and Technology

^{††} 名古屋大学統合音響情報研究拠点/情報メディア教育センター
Center for Integrated Acoustic Information Research/
Center for Information Media Studies, Nagoya University

^{†††} 名古屋大学統合音響情報研究拠点/工学研究科
Center for Integrated Acoustic Information Research/
Graduate School of Engineering, Nagoya University

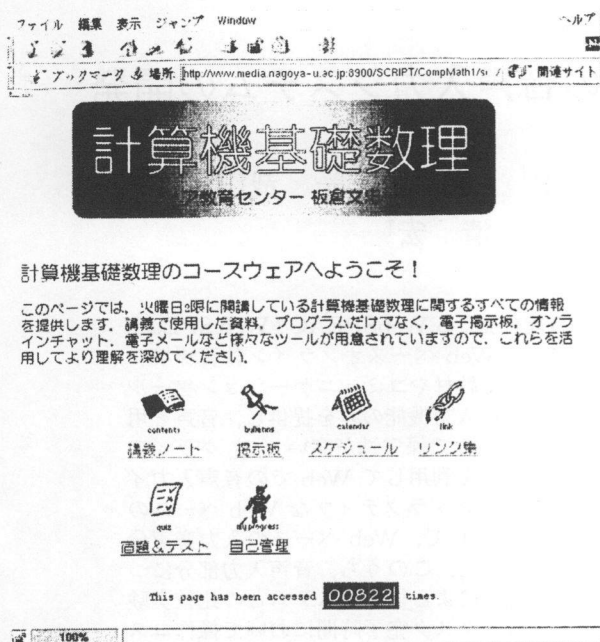


図 1 名古屋大学での WebCT の利用例
 Fig. 1 Example of WebCT at Nagoya Univ.

境 WebCT (World Wide Web Course Tools)^{1)~3)}を用いた教育用オンラインシステムによる講義を試みている(図1)。コースウェアとは、計算機システム上に作られたオンライン教材のことであり、WebCTは、インターネットの主要アプリケーションであるWorld Wide Web (WWW)をベースに、マルチメディアコースウェアに加えて生徒間や生徒と教師間のコミュニケーションツールや試験など講義において必要な機能を含めたオンライン教育環境を作成するためのソフトウェアである。すでにアメリカ合衆国やカナダを中心に57カ国1528(平成12年9月現在)の大学などでの使用実績があり、最近では大手出版社によりWebCT上で動作する教材データの販売が始まっている。このようにWebCTは、これまで標準的なプラットフォームのなかった教育用オンラインシステムの実用上の標準となりうるシステムである。

WebCTでは、コースウェアの中で利用できる様々なコンテンツが部品として提供される。しかし、これまでWebCTでは音声出力を持つ部品の提供はできたが、音声入力の機能を持つ部品の提供はなかった。これはWebCTがベースとしているWebに手軽に利用できる音声入力の機能がなかったためである。

音声入力機能をコースウェアに加えることによって、より学習効果の高いマルチメディアコンテンツとクラス内での柔軟なコミュニケーションなどの提供が可能になる。たとえば、入力音声を受講者が信号処理技術により自ら分析することのできるコースウェアは音声信号処理の学習の際に効果がある。また、音声認識エ

ンジンと組み合わせることによって音声による高度なインタフェースの実現も可能である。さらに、遠隔教育システムでは、地理的に離れている場合や同一時間にいない受講者同士や受講者と教師間のコミュニケーション機能の充実にも使用可能である。これまでのシステムでは、この機能をテキストベースの掲示板やチャットによって実現してきたが、テキストベースのシステムでは、話の微妙なニュアンスが伝わりにくく、伝えたい内容を正確に伝えるのは難しい。そこで、音声入力機能を加えることにより、テキストベースのものに加えて音声ベースの掲示板やチャットが実現可能になり、この問題を解決できる。ほかにも遠隔教育における試験では、受講者が本当に受講者本人なのかを音声によって認証するシステムを本システムによって実現できる。このような認証には指紋認証による方法も考えられるが、指紋を読み取るためには専用のハードウェアを別途用意しなければならない。音声による認証では、計算機に標準的に備わっている音声入力ハードウェアのみで、低コストの認証システムを実現できる。

そこで、本研究では、実際にWebCTのようなWebベースコースウェアに音声入力機能を加えるシステムであるWebSPEAC (Web SPEech Acquisition for Courseware) Systemを作成した。また、本システムを利用して、教育現場で利用可能なアプリケーションを作成し、実際のユーザを被験者にした実験によって評価を行った。

本論文の構成は以下のとおりである。2章では、WebSPEAC Systemの概要について述べる。3章では、本システムの構成について述べる。4章では、本システムの実装およびアプリケーションの例について述べる。5章では、本システムの有効性を示すために行った評価実験について述べる。最後に6章で本論文のまとめを行う。

2. WebSPEAC System の概要

WebSPEAC Systemは、マルチメディアコースウェアに音声入力機能を加えるためのシステムである。主にWebCTの中で利用することを前提として開発しているが、WebCTはWebをベースにしたシステムため、一般的なWebシステム上でも動作する。また、本システムは、(1) Webブラウザには音声入力機能のみを提供し、音声を用いるアプリケーションプログラムはWebサーバ上で一括管理するので保守性が高い、(2) クライアント側は、簡単な初期設定のみで利用できる、(3) サーバプッシュを利用してWeb環境での音

声入力インタフェースを実現している、という特徴を持つ。

本システムでは、まず Web ブラウザ側では音声の録音のみを行い、その音声データを Web サーバ側に送信する。次に音声を受け取ったサーバ側で音声の加工や処理を必要に応じて行う。その結果を基に Web コンテンツを生成し、ブラウザに送り出すことによって、音声入力機能を持った Web アプリケーションを実現する。

この方法では、クライアント側で音声の加工や様々な処理を行う方法と比べた場合、音声を加工処理するアプリケーションプログラムやデータベースなどのリソースがサーバ側で一括管理できるという特徴がある。また、音声処理プログラムの更新があった場合もサーバ側のプログラムを置き換えるだけで済むので端末の多い計算機システムや家庭などでの個人ユーザの利用においても保守性が高いのが利点である。

また、従来からも本システムと同様のシステムをファイルアップロード⁴⁾やメール⁵⁾などを音声送信の手段として利用することで構築することは可能であった。しかし、サウンドレコーダ(録音用ソフトウェア)などを用いて音声をあらかじめファイルに保存しておき、その音声データをマウスのクリックやキーボードの入力などの操作によって送信する作業(ファイルアップロード作業)が必要であった。本システムでは、ユーザはマウスやキーボード操作を用いたファイルの送信などを意識することなく利用できる音声入力インタフェースを利用することができる。

設定に関しては、音声入力機能を加えるためのヘルパアプリケーションを Web ブラウザに初期設定で登録だけで、クライアント計算機に複雑で大規模な信号処理プログラムなどをインストールする必要がないのも利点である。

一方で、本システムは、Web サーバ上で一括して音声の処理をするため、負荷がサーバに集中するというデメリットが存在する。このため Web サーバには高機能な計算機を利用する必要がある。また、ネットワークを介して処理を行うため、現状では高容量データや実時間性を必要とするアプリケーションには適していないのもデメリットである。しかし、これら問題は、計算機の高速化やネットワークの広帯域化によって次第に解消される問題であると考えている。

3. WebSPEAC System の構成

3.1 システムの全体構成

本システムの構成図を図 2 に示す。本システムは、

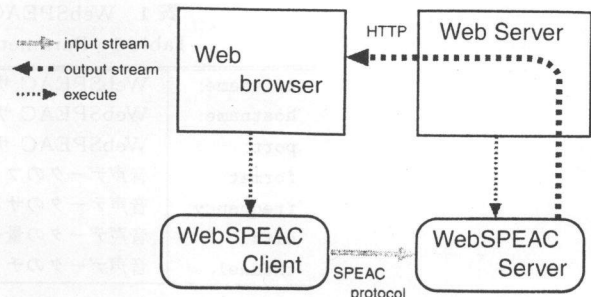


図 2 WebSPEAC システム構成図

Fig. 2 Components of WebSPEAC System.

WebSPEAC サーバを中心とした複数のプログラム群とそれらを結ぶ TCP/IP 上のプロトコルで構成する。プログラム群は、ローカル側の計算機上で使用するクライアント側プログラムと Web サーバ側プログラムに分けることができる。クライアント側プログラムは、Web ブラウザと Web ブラウザに音声入力機能を加えるためのヘルパアプリケーションである WebSPEAC クライアントである。Web サーバ側プログラムは、Web サーバと WebSPEAC サーバである。Web ブラウザと Web ブラウザは、一般に普及しているものを利用する。また、クライアント側プログラムと Web サーバ側プログラム間のプロトコルを WebSPEAC プロトコルと呼ぶ。

3.2 WebSPEAC サーバ

WebSPEAC サーバは、Web サーバの中で実行される CGI (Common Gateway Interface) プログラムである。ユーザが、Web ブラウザを用いて本システムを用いたスタートページにアクセスすると、Web サーバによって WebSPEAC サーバは起動される。WebSPEAC サーバは、起動後、Web ブラウザに対して音声録音要求を行う。要求を受けた Web ブラウザは、WebSPEAC クライアントを起動して、音声の録音を行う(図 3)。

録音要求を送信した後、WebSPEAC サーバは、WebSPEAC クライアントからの録音音声の受信が完了するまで処理を中断して待機する。受信完了後、音声を加工や処理する音声処理モジュールを用いて受信した音声を加工する。最後にその結果を Web ブラウザに送り出して、処理を終了する。

以上のように WebSPEAC サーバが本システムの全体の処理の流れを制御するメインプログラムであり、音声処理モジュールをユーザの目的のあったものに変えることによって、様々な WebSPEAC アプリケーションを構成できる。

3.3 WebSPEAC プロトコル

WebSPEAC プロトコルは、WebSPEAC クライア

表 1 WebSPEAC プロトコルのパラメータの例
Table 1 Parameters of WebSPEAC protocol.

filename:	WebSPEAC サーバが音声受信時に使用するファイル名
hostname:	WebSPEAC サーバが動作するホスト名 (Web サーバ)
port:	WebSPEAC サーバが音声受信時に使用するポート番号
format:	音声データのフォーマット形式 (default: WAV 形式)
frequency:	音声データのサンプリング周波数 [Hz] (default: 16 kHz)
bits:	音声データの量子化ビット数 [8 or 16]
channel:	音声データのチャンネル数 [1 or 2]

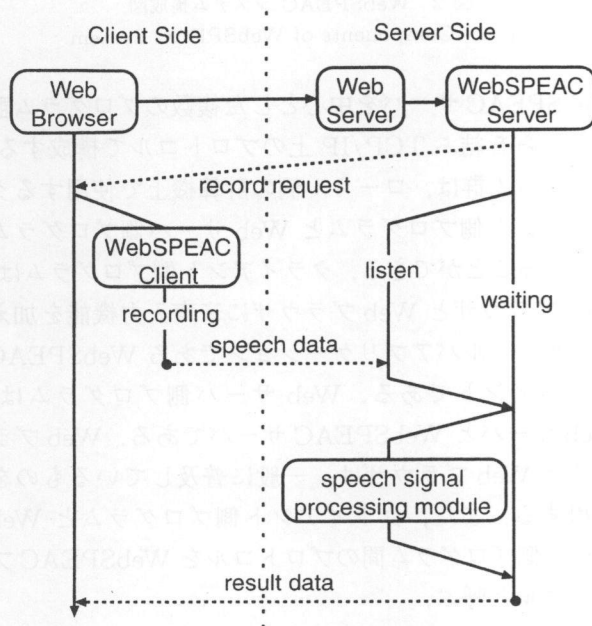


図 3 WebSPEAC システム動作図

Fig. 3 Communication chart of WebSPEAC System.

ントと WebSPEAC サーバ間の通信を行うプロトコルの総称である。HTTP (Hyper Text Transfer Protocol)⁶⁾ 上のメッセージおよび TCP/IP のソケット間通信で実装する 2つのプロトコルがあり、それぞれが録音開始要求と音声データ送信の機能を担っている。

“録音開始要求”は、WebSPEAC サーバから Web ブラウザ側に送信されるプロトコルであり、後述するマルチパートドキュメントによる HTTP メッセージ中の論理パートの 1つとして記述する。

“録音開始要求”を受信した Web ブラウザは、WebSPEAC クライアントによって音声の録音を開始する。また、録音開始要求と同時に、そのメッセージの中には、サンプリング周波数などの録音時の条件を WebSPEAC クライアントに渡すパラメータを記述できる。記述できるパラメータの例を表 1 に示す。これらパラメータを利用することで、音声の符号化方法などの指定が可能である。また、これらの中で必須のパラメータが filename, hostname, port である。この 3つは、後述するように録音後の音声データ送信時において重要な意味を持つパラメータである。

マルチパートドキュメント⁷⁾とは、HTTP によるアクセスに対して返されるメッセージ中に MIME (Multipurpose InternetMail Extensions)⁸⁾によって分けられた論理上の複数パートを持たせる技術である。そして、この論理パートを Web サーバ側の都合のよいときに生成することで、Web ブラウザへの出力を Web サーバ側から送り出す技術がサーバプッシュである。

本システムを実現するには、サーバ側で生成した Web コンテンツを Web ブラウザに任意のタイミングで送信する必要がある。しかし、Web コンテンツの送信に使う HTTP は、Web ブラウザから Web サーバへリクエストに対してレスポンスを返すように設計されている。このため、Web サーバ側で Web ブラウザの内容を制御することはできない。本システムでは、これをサーバプッシュを利用することで解決している。つまり、WebSPEAC サーバは、音声の受信や音声処理モジュールの動作と同期をとりながら、図 4 の例で示すようなマルチパートドキュメントを生成している。

図 4 の例では、マルチパートドキュメントごとに固有な区切り文字列 (例では “--WebSPEAC918810625”) で区切った 4~11 行 (第 1 パート)、11~18 行 (第 2 パート)、18~25 行 (第 3 パート) が、それぞれ論理パートである。各パートは、最初の “Content-type:” で始まる行により、そのパートの中身の論理的意味を明示する。この例では、第 1, 3 パートが HTML (Hyper Text Markup Language)⁹⁾ であり、第 2 パートは、WebSPEAC プロトコルの Web サーバへの録音開始要求である。HTML パートの中身は、具体的には、第 1 パートが音声入力をユーザに促すための画面構成を記述した HTML であり、第 3 パートには最終処理結果を記述するための HTML を記述する。

WebSPEAC クライアントが録音した音声データは、TCP/IP のソケット間通信で実装するもう 1つの WebSPEAC プロトコルである “音声データ送信” によって Web サーバ側に送信する。図 5 に音声データ送信の動作図を示す。

WebSPEAC サーバは、WebSPEAC クライアント

```

1: HTTP/1.0 200 OK
2: Content-type: multipart/x-mixed-replace;
   boundary=WebSPEAC918810625
3:
4: --WebSPEAC918810625
5: Content-type: text/html
6:
7: <html>
8: .....
9: </html>
10:
11: --WebSPEAC918810625
12: Content-type: application/x-WebSPEAC
13:
14: filename: fw3228.wav
15: hostname: fw
16: port: 30352
17:
18: --WebSPEAC918810625
19: Content-type: text/html
20:
21: <html>
22: .....
23: </html>
24:
25: --WebSPEAC918810625--

```

図4 マルチパートドキュメントの例
Fig. 4 Example of multipart document.

からの音声の送信が開始するまで，“録音開始要求”のときに与えるパラメータのうち、hostname と port で構成されるソケットに対して受信可能状態 (listen) のまま待機する。このとき、WebSPEAC サーバ全体は、処理を中断した状態である。

WebSPEAC クライアントは、音声の録音後、WebSPEAC サーバに対して接続を試みる。接続確立後に、WebSPEAC クライアントは、文字列 filename を送信する。WebSPEAC サーバは、filename が自分で処理すべきファイル名であるかを判断し、正しければデータの受信を開始する。自分が受信するべきではない filename の場合、WebSPEAC クライアントに対して、接続の拒否を通知して、再び listen 状態での待機を再開する。

ここで、hostname、port、filename は、Web ブラウザから WebSPEAC サーバへのアクセスごとに一意に定められた組合せである。WebSPEAC サーバは、起動時に使用可能な空きソケットの割当てを行い、

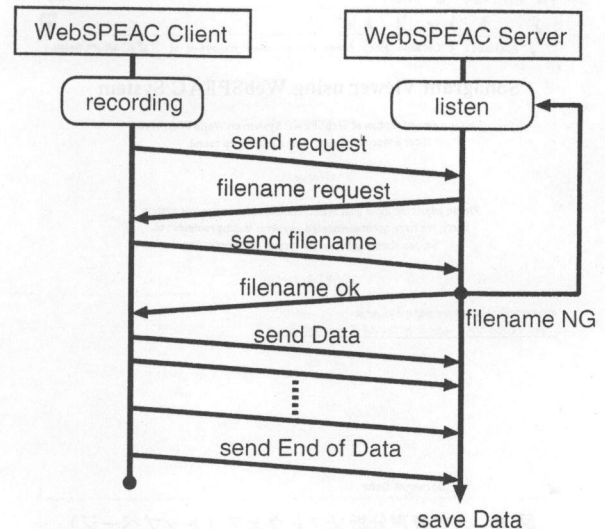


図5 WebSPEAC クライアントとサーバ間の通信
Fig. 5 Communication chart of WebSPEAC protocol.

“録音開始要求”のパラメータの中で WebSPEAC クライアントに、そのパラメータの通知を行う。この一連の動作の中で filename は、接続時の認証のための一種のパスワードとしての役割をする。

なお、本システムを利用した Web アプリケーションへのアクセス認証は Web サーバの認証システムを利用することになるため、本システムの中では特別なプロトコルなどは用意していない。

4. 実装方法およびアプリケーションの例

4.1 実装方法および動作環境

今回、開発および動作環境としてパーソナルコンピュータ (PC) 上の Linux を用いた。開発言語には、WebSPEAC サーバは Perl、WebSPEAC クライアントには C 言語を用いた。現在、WebSPEAC クライアントの動作を確認している OS は Linux のみである。これは WebSPEAC クライアントが PC のサウンドデバイスを用いて録音を行うが、OS ごとにサウンドデバイスを扱うプログラミング方法が異なるためであり、他の OS への移植も予定している。WebSPEAC サーバは、Web サーバ上で動作する CGI プログラムなので、多くの UNIX 系の OS で動作する Web サーバで動作する。

ユーザが利用する Web ブラウザには、マルチパートドキュメントをサポートしたものである必要がある。この機能をサポートし動作確認をしているものとしては、Netscape 社の Netscape Navigator がある。この機能は、RFC (Request for Comments)⁶⁾ で定義されており、RFC2616 に準拠したブラウザでは動作する。

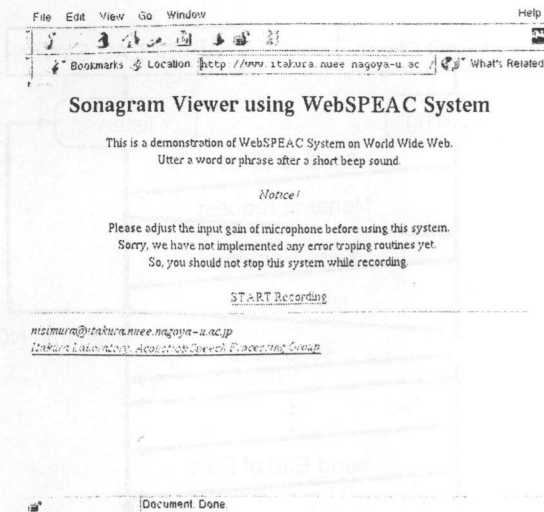


図 6 入力音声分析ソフトウェア (トップページ)
Fig. 6 Snapshot of Speech Analyzer (1).

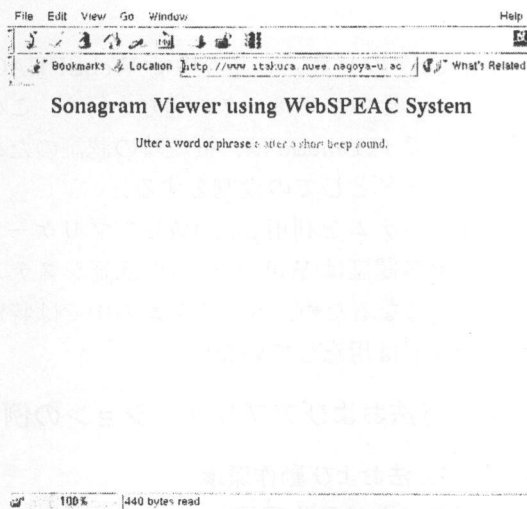


図 7 入力音声分析ソフトウェア (音声入力中)
Fig. 7 Snapshot of Speech Analyzer (2).

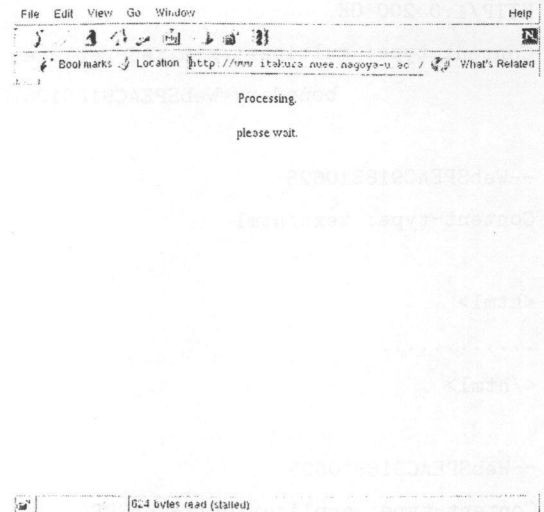


図 8 入力音声分析ソフトウェア (データ送信および加工処理中)
Fig. 8 Snapshot of Speech Analyzer (3).

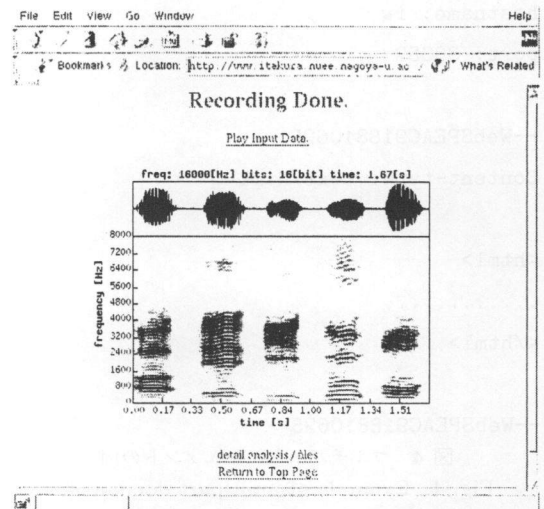


図 9 入力音声分析ソフトウェア (分析結果表示)
Fig. 9 Snapshot of Speech Analyzer (4).

また、本システムは、処理中の音声データをディスク上に蓄えるため、録音時間はシステムのリソースにより制限される。

4.2 アプリケーションの例

本システムを用いて、実際の教育現場において利用可能なアプリケーションの開発を行った。以下にアプリケーションの例を3つあげる。

4.2.1 入力音声分析ソフトウェア

デジタル信号処理の講義での利用を想定した音声の簡易分析システムを作成した。このソフトウェアは、Web上から音声の入力を行い、入力音声の波形とサウンドスペクトログラムを表示する。また、時間で区間を指定することによる波形の切り出しや高域強調のパラメータの変更などの再分析もできる。

このソフトウェアを利用することで学習者が入力した自分の声を実際に分析できる。これは音声信号処理

の学習において、学習効果の高いコンテンツになる。本ソフトウェアの実行画面を図6～図9に示す。

4.2.2 音声認識システム

前述のように本システムは、音声認識プログラムと組み合わせることもできる。今回は、大語彙連続音声認識プログラム Julius¹⁰⁾と組み合わせることにより、入力された音声を生声認識し、認識結果を表示するソフトウェアを作成した。今回は認識結果をWebブラウザ上に出力するのみだが、認識結果をテキスト処理プログラムで加工することで、音声認識による語句検索システムなどに応用できる。

本ソフトウェアの実行画面を図10に示す。

4.2.3 受験者認証システム

遠隔教育での試験を想定した音声による受験者認証システムを作成した。ただし、現状では自動的に話者を識別することはせず、試験官が試験の回答に添付き

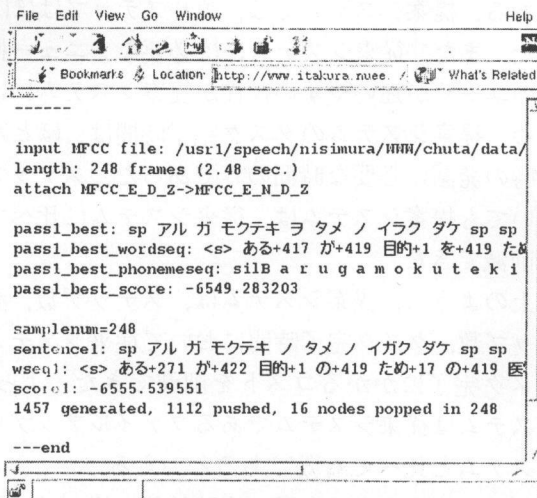


図 10 音声認識システム動作画面（認識結果を表示）
Fig. 10 Snapshot of Speech Recognition System.

れた受験者の入力音声とあらかじめ登録された受験者の声と聞き比べて判断する。また、認証時に使う音声パスワードは、あらかじめ録音された音声を使われる不正を防ぐため、認証時にランダムに決定する。

5. 評価実験

5.1 実験方法

実験では、4.2.1 項で述べた「入力音声分析ソフトウェア」を本システムと従来法であるファイルアップロードシステムによって作成し、それぞれをユーザに使用してもらうことで比較した。

被験者は、本システムのユーザとして想定している 20 代男女である。また、計算機の使用歴も結果に影響を与える可能性があるため、被験者を計算機の初心者 (beginner) と熟練者 (expert) に分け、それぞれ 8 名とした。初心者と熟練者のグループ分けは、初心者の基準を計算機の使用歴 1 年未満程度として、被験者自身による判断によって行った。

評価尺度には、タスク完了までの音声の入力、マウスのクリック、キーボードによる文字列の入力をそれぞれ 1 ステップとした (1) ステップ数, (2) 誤りステップ数, (3) タスク完了に必要な時間 (タスク完了時間) を用いた。ここでタスクの完了とは、本システムを利用した Web のトップページにアクセスしてから、音声を録音して、WebSPEAC サーバに送信完了することとし、送信後のシステムの処理の時間は含まない。

被験者がシステムに不慣れな場合、タスクを完了できない場合もある。これを防ぐために従来システム、提案システムともに簡単なマニュアルを用意した。実験前にマニュアルを読むことで、被験者が実験の手順

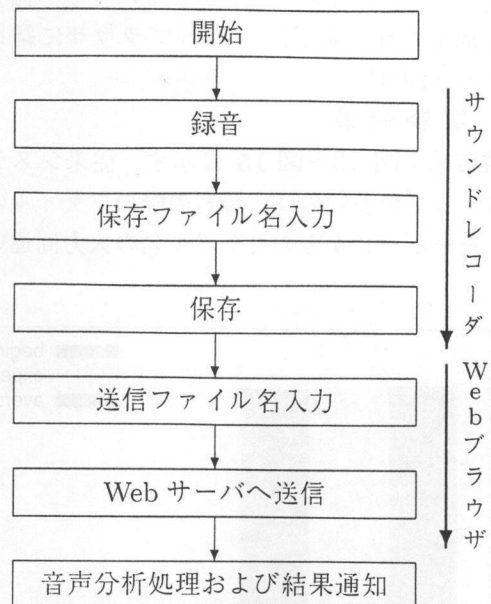


図 11 従来システム操作手順
Fig. 11 Steps of file upload system.

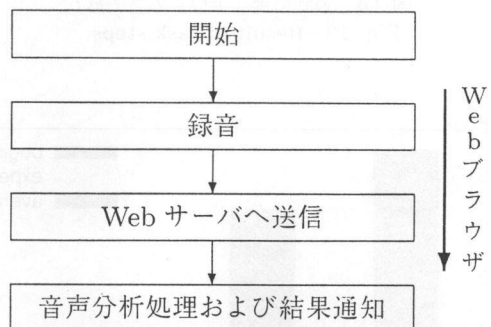


図 12 提案システム操作手順
Fig. 12 Steps of WebSPEAC system.

を前もって把握できるように心掛けた。以下にそれぞれのシステムの具体的な操作手順について述べる。

従来システムでは、まずサウンドレコーダを用いて音声の録音を行い、その音声をファイルに保存する。次に Web ブラウザの送信ファイル名フォームに先ほどの音声ファイルのファイル名を入力し、分析開始 (ファイル送信) ボタンをクリックすることによってサーバへの送信を行う。図 11 に従来システムの手順を示す。

提案システムにおいては、トップページの開始ボタンをクリックすることによってプロセスを開始すると、WebSPEAC クライアントが録音を行い、録音完了後、自動的に音声データをサーバへ送信する。また、録音時の発話のタイミングを被験者に知らせるために録音開始時に発信音を鳴らす。図 12 に提案システムの手順を示す。図 12 中の各ステップの動作例は、「開始」が図 6、「録音」が図 7、「Web サーバへ送信」が図 8、「音声分析処理および結果通知」が図 9 に相当する。

従来システム、提案システムともにサーバ側でのすべての加工処理が完了後、Webブラウザに結果が表示される(図9)。

5.2 実験結果

実験結果を図13~図15に示す。従来システムに誤りステップが多いのは、録音の際のファイルの保存忘れやキーボードからのファイル名の入力間違いのためである。

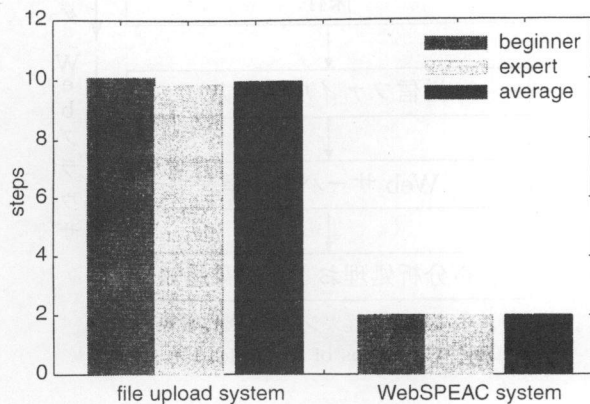


図13 実験結果(平均ステップ数)

Fig. 13 Result of task steps.

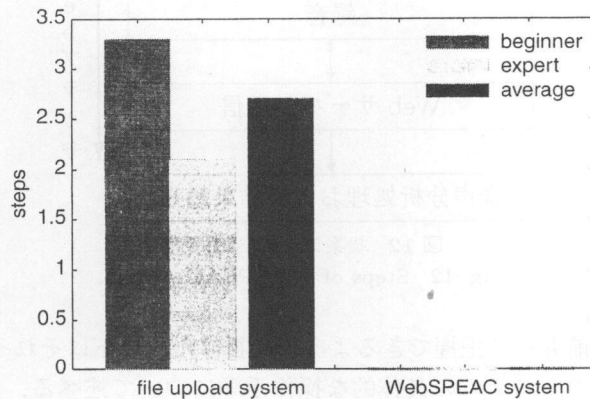


図14 実験結果(平均誤りステップ数)

Fig. 14 Result of error steps.

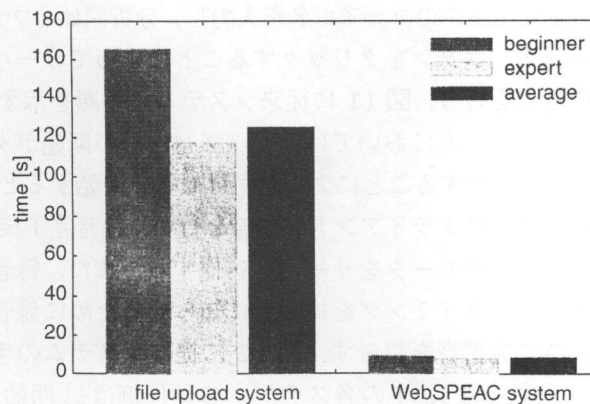


図15 実験結果(平均タスク完了時間)

Fig. 15 Result of elapsed time.

めである。提案システムでは、誤りステップはほぼなかった。また、従来システムには必要なメニュー操作などのユーザが迷いやすい操作が提案システムにはないため、提案システムのタスク完了時間は、ほとんど録音時の発話に必要な時間のみであった。ステップ数においても提案システムは、従来システムに比べて大幅に削減できた。

以上のように、提案システムは、ステップ数、誤りステップ数、タスク完了時間において従来システムよりタスク完了にかかるコストを削減できた。よって、本システムは従来システムであるファイルアップロードシステムと比べて有効であるといえる。

初心者と習熟者の両者間の実験結果の違いは、習熟者の方がステップ数、誤りステップ数、タスク完了時間のすべてにおいて多少良い結果になる程度であり、実験結果の傾向に違いはなく、本システムは両者において有効であるといえる。

ただし、提案システムは、一度プロセスをスタートすると被験者が途中で止めることができないのが不便という意見や録音時の発話のタイミングが、信号音だけでは分かりにくいという意見もあったので、今後の改良が必要であることも分かった。

6. むすび

本論文では、WebをベースにしたWebCTによるマルチメディアコースウェアに音声入力を機能を加えるWebSPEAC Systemについて述べた。本システムにより、コースウェアに音声入力機能を持つマルチメディア教材や音声によるコミュニケーション機能などを加えることができる。その応用としては、音声分析システムや音声認識アプリケーション、そして、遠隔教育における試験の受験者認証システムなどがある。

また、評価実験を通じて、本システムは従来法であるファイルアップロードを利用したシステムに比べて有効であることを示した。

今後は、システムの改良とともに、本システムを実際に組み込んだコースウェアの開発をする必要がある。そして、本システムの実際の教育現場での学習への有効性や問題点の検討をする必要がある。

参考文献

- 1) Goldberg, M.W. and Salari, S.: An Update on WebCT (World-Wide-Web Course Tools) - A Tool for the Creation of Sophisticated Web-Based Learning Environments., *Proc. NAUWeb '97* (1997).

- 2) WebCT Inc.: WebCT.
<http://www.webct.com/>.
- 3) 梶田, 板倉: WebCT によるコースウェア作成支援環境の構築, 電子情報通信学会技術研究報告, Vol.ET99-58, pp.15-22 (1999).
- 4) Nebel, E. and Masinter, L.: Form-based File Upload in HTML, RFC1867 (1995).
- 5) 熊谷, 三輪: 電子メールと Web 技術を用いた公衆型音声分析システム, 日本音響学会春季講演論文集, pp.325-326 (1999).
- 6) Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P. and Berners-Lee, T.: Hypertext Transfer Protocol - HTTP/1.1, RFC2616 (1999).
- 7) Netscape Communications Corp.: An Exploration of Dynamic Documents.
http://home.netscape.com/assist/net_sites/pushpull.html.
- 8) Freed, N., Borenstein, N., Moore, K., Klensin, J. and Postel, J.: Multipurpose Internet Mail Extensions (MIME), RFC2045, RFC2046, RFC2047, RFC2048, RFC2049 (1996).
- 9) World Wide Web Consortium (W3C): Hyper Text Markup Language.
<http://www.w3.org/MarkUp/>.
- 10) 武田, 河原, 伊藤, 鹿野: 日本語ディクテーション基本ソフトウェア, 情報処理学会全国大会講演論文集, Vol.2, pp.461-462 (1998).

(平成 12 年 3 月 17 日受付)

(平成 12 年 12 月 1 日採録)

西村 竜一 (学生会員)



1999 年名古屋大学工学部電気電子情報工学科卒業。同年, 奈良先端科学技術大学院大学情報科学研究科博士前期課程に入学, 現在に至る。主に音声認識を中心とする音声インタフェースの可能性に関心を持っている。また, オンライン教育システムの開発にも興味がある。

梶田 将司 (正会員)



1990 年名古屋大学工学部情報工学科卒業, 1995 年同大学院博士後期課程修了。名古屋大学情報メディア教育センター助手, 工学博士。日本音響学会, 電子情報通信学会, アメリカ音響学会, IEEE 各会員。

武田 一哉 (正会員)



1983 年名古屋大学工学部電気工学科卒業, 1985 年同大学院博士前期課程修了。同年国際電信電話 (株) 入社, ATR 自動翻訳電話研究所, KDD 研究所において音声合成・認識システムの研究を行う。1994 年名古屋大学工学部助教授, 工学博士。IEEE, 日本音響学会, 電子情報通信学会各会員。

板倉 文忠



1963 年名古屋大学工学部電子工学科卒業。1968 年同大学院博士課程修了。同年電電公社 (現 NTT) 武蔵野通研入所。音声処理の研究に従事。工学博士。1973~1975 年ベル研究所にて音声認識・音声分析の研究を行う。1984 年名古屋大学工学部教授。昭和 45, 53, 56 年度論文賞, 昭和 47, 56 年度業績賞受賞, IEEE, 日本音響学会各会員。

鹿野 清宏 (正会員)



1970 年名古屋大学工学部電気工学科卒業。1972 年同大学院修士課程修了。同年電電公社武蔵野電気通信研究所入所。1984~1986 年カーネギーメロン大客員研究員。1986~1990 年 ATR 自動翻訳電話研究所音声情報処理研究室長。1992 年 NTT ヒューマンインタフェース研究所主席研究員。1994 年より奈良先端科学技術大学院大学情報科学研究科教授。音情報処理学講座を担当。工学博士。主として音声・音情報処理の研究および研究指導に従事。昭和 50 年電子通信学会米沢賞。平成 3 年 IEEE SP 1990 Senior Award, 平成 6 年日本音響学会技術開発賞。平成 12 年情報処理学会山下記念研究賞。IEEE, ISCA, 音響学会, 電子情報通信学会各会員。