自然言語処理 119－1
音声言語情報処理 16－7
（1997．5．26）

WWWブラウザの音声による制御

渕 武志　加藤 恒昭
ＮＴＴ情報通信研究所

抄録

既存のWWWブラウザに対し、音声によってJavaScriptコードを実行させる技術を開発した。Webページ上に特別なアプレットを配置し、そのパラメータとして、認識させる言葉の読み仮名と、その言葉が認識された場合に起動されるJavaScriptの関数を記述することで、ブラウザを音声で制御できるようにした。さらに、プロキシによって転送中のＨＴＭＬ文書を解析し、このアプレットの呼び出し記述を適切なパラメータと共に当のHTML文書に自動的に追加することにより、任意のHTML文書に対して音声による制御を可能にした。

Voice Controllable WWW Browser

Takeshi Fuchi, Tsuneaki Kato
NTT Information and Communication Systems Laboratories

Abstract

We have developed a technology for controlling an existing WWW browser by voice. To make a WWW page voice-sensitive, you have only to add a special Applet and the list of word/JavaScript code combinations to the source of the page. Any JavaScript codes can be activated in response to voice, so the browser can be controlled by voice freely. In addition, by analyzing a WWW page in our proxy, our system automatically attaches the Applet and word/JavaScript code combinations to the source of the page. This allows ordinary WWW pages without information for voice recognition to get sensitive to voice.

1　はじめに

World Wide Webの普及により、コンピュータ及びコンピュータネットワークを利用する人口が急速に拡大している。そしてWWWの窓口たるWWWブラウザには様々な拡張がなされつつある。そのような拡張の一つとして、Webページが音声に反応できるようにすることが挙げられる。コンピュータを音声によって制御するシステムは既にApple社[1]やIBM社[2]などから提供されている。しかし、これらのシステムは基本的にあらかじめ登録された単語に対して、ファイルの複製やウィンドウの開閉などの固定的な機能を結びつけるものである。従って、そのままではWWWブラウザという特定のアプリケーションを細かく制御することはできない。これに対して、DigitalDreams社[3]からブラウザを音声によって制御するためのシステムが提供されている。これはApple社の音声認識ライブラリを

利用して、ブラウザの細かい制御を可能にするシステムである。このシステムは、ブラウザのメニューに登録されている機能を音声によって実行する機能のほかに、表示されているリンクをユーザが発声することで、ブラウザにリンク先のWebページへ表示を切り替えさせる機能を提供している。この機能は、ブラウザが表示中のWebページの内容を解析し、ページの内容に応じて認識対象となる単語を動的に変えることで実現されている。しかし、それらの単語の認識によって起動される動作は、リンク先への表示の変更という機能に固定されている。他に、土肥らの開発したシステム[4][5]では、WWWブラウザは各Webページに対してあらかじめ登録した言葉による指示を受け付ける。登録のないWebページに対しては、リンクのリストを提示し、リンクに結びつけられた番号を発声することでそのリンクへ表示を切り替える機能を提供している。これも、音声に結びつけられるブラウザの動作はリンク先への表示の変更のみである。一方、中嶋らの開発したシステム[6]では、対象の指定はマウスを用いるものの、ブラウザの動作を音声で指示できる。しかし、指示できる動作は、ファイルの読み上げやダウンロードなど、あらかじめ用意されたものに限られる。

我々は上記のような問題点を解決するための仕組みを考案し、実装した。本稿ではそのシステムの概要について報告する。次節で我々のシステムの特徴を述べる。3節では音声によってJavaScriptコードを起動する仕組みの概要と、音声制御のためのWebページの記述例について述べる。4節では、HTML文書を解析するプロキシの概要と、その応用例について述べる。5節では、残された課題と今後の展望について述べる。

2　システムの特徴

従来のシステムの問題点を解決するため、我々は以下のような特徴を持つシステムを開発した。

(1) 音声によって、WWWブラウザにJavaScriptコードを実行させることができる。

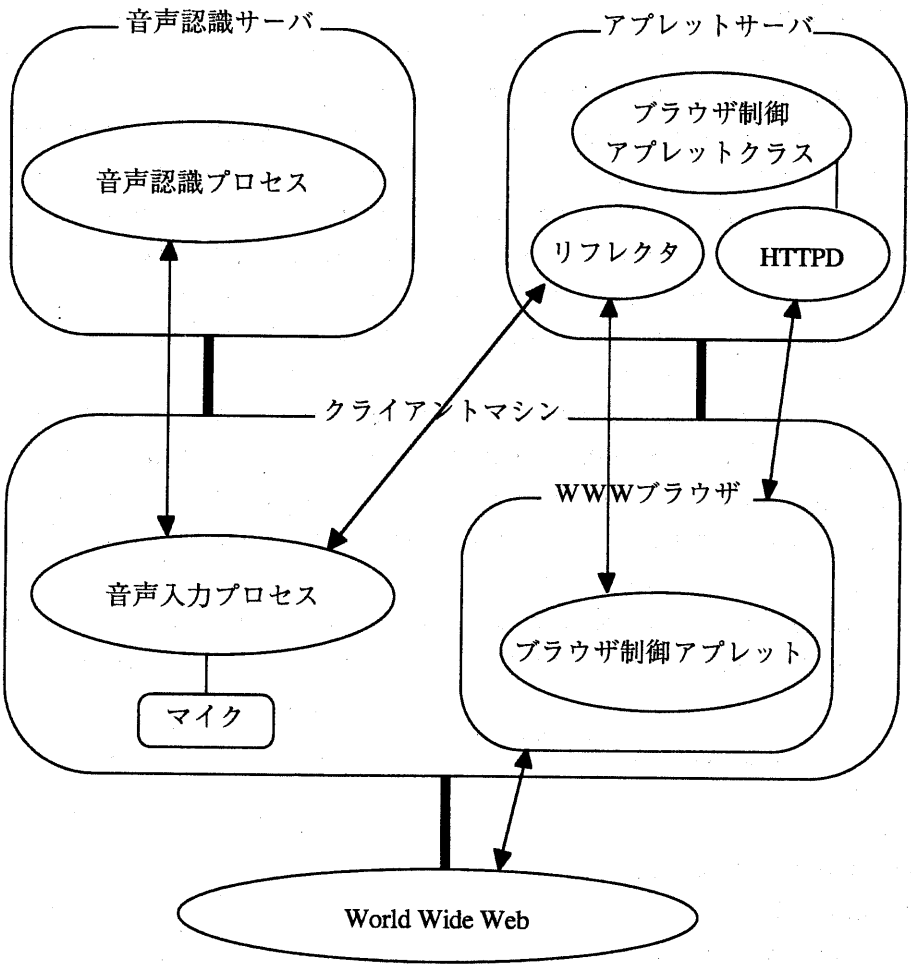
(2) 実行させるJavaScriptコードは、Webページのソースファイル内で指定する。

(3) 音声制御用の情報を含まない一般のWebページに対しても、プロキシによって情報を追加することにより、ブラウザの音声制御を可能にする。

(4) 日本語のリンクに含まれる単語の発声によって、ブラウザの表示をリンク先のページに切り替えさせることができる。

我々のシステムでは、必要な情報をWebページ中に記述するようになっているため、Webページの提供者がページ毎に簡単に音声制御の仕方を設定できる。これによってJavaScriptの機能の範囲内で任意の動作を音声によって起動させることができ、様々なサービスへの応用が容易にできるようになる。但し、この機能のみだと、あらかじめ音声制御用の情報を埋め込んだWebページをブラウザが表示中である場合にしか、音声による制御をすることができない。そのため、プロキシによってHTML文書の転送中にページの内容を解析して、音声制御用の情報を自動的にHTML文書に埋め込むようにしてある。これによって、任意のWebページに対して音声制御をすることが可能となる。但し、この場合には、音声によるリンクの指示やページの読み上げなど、プロキシで設定する既定の機能に限られる。その他、プロキシにおいて日本語形態素解析システム[7]を用いてリンク部分を解析することにより、リンクを示す文字列に含まれる漢字に読み仮名を振り、これによって、漢字を含むリンクに対しても音声による指示を可能にしている。

図1　音声制御部の構成



3　ブラウザ音声制御部

音声制御部の構成を図1に示す。システムはクライアントマシン、アプレットサーバ、音声認識サーバの三つのマシンで構成されている。音声認識サーバで動いている音声認識プロセス[1][8]は、認識単語候補と音声データを受け取ると、認識単語候補の中から音声データに最も近い単語を返すプロセスである。アプレットサーバでは、HTTP デーモンが動いており、要求があるとブラウザ制御アプレットを供給する。アプレットサーバでは他にリフレクタと呼ぶプロセスが動作している。これは、他のマシンからソケットで接続されると、そのマシンに対して別のソケットでもう一つの通信路を作り、これらの二つの通信路の間でデータの送受信を中継するプロセスである。このプロセスは、アプレットにおけるセキュリティのためのアクセス制限の中で、アプレットがクライアントマシン上のプロセスと通信できるようにするために用いられる。クライアントマシンではWWWブラウザと音声入力プ

[1] 本研究に際し、NTTヒューマンインターフェース研究所音声情報研究部で開発した音声認識サーバシステムECLAIRを使用しました。

ロセスが動いている。そして、WWW ブラウザ上ではブラウザ制御アプレットが動いてる。このアプレットは、認識単語候補を音声入力プロセスに通知し、そこからの返答に応じて JavaScript のコードを実行する。音声入力プロセスは、マイクから入力があると認識単語候補と音声データを音声認識サーバに送り、そこからの答えをブラウザ制御アプレットに返す。

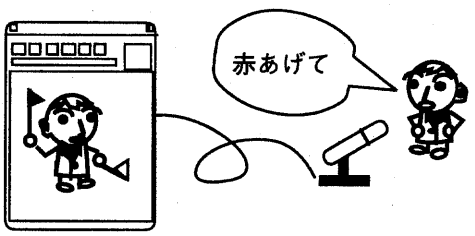
以下に、システムの動作の様子を記す。まず、ユーザがブラウザ制御アプレットの呼び出しの記述を含む HTML 文書をブラウザで表示する。するとブラウザはアプレットサーバにアプレットを要求し、送られてくるブラウザ制御アプレットを実行する。ブラウザ制御アプレットは、HTML 文書に記述されたパラメータの中から認識単語候補と、それらの単語が認識された際に実行する JavaScript のコードを読みだす。次にこのアプレットはアプレットサーバ上のリフレクターにソケットを張る。リフレクタはソケットを張られると、ソケットを張ってきたマシン上の音声入力プロセスにさらにソケットを張り、これらのソケット間の中継を開始する。この時点で、ブラウザ上のブラウザ制御アプレットと音声入力プロセスとの間の通信路ができたことになる。通信路が確立すると、ブラウザ制御アプレットは音声入力プロセスに対して、認識単語候補を送る。音声入力プロセスは認識単語候補を受け取るとマイクの監視を始める。マイクからの入力があると、これを音声データとして記録する。そして、音声認識サーバとの通信を開始し、認識単語候補と音声データを送信する。音声認識サーバは認識単語候補の中から音声データに最も近い単語を算出し、この単語を音声入力プロセスに返す。音声入力プロセスは、音声認識サーバから返された単語をブラウザ制御アプレットに返す。ブラウザ制御アプレットは、単語を返されると、その単語と結びつけられた JavaScript のコードを実行する。

図2　ブラウザ制御アプレット呼び出しの例

```
<HTML>
<HEAD>
<SCRIPT LANGUAGE="JavaScript">
function change(x) {
...the JavaScript code of change(x)...
}
</SCRIPT>
</HEAD>
<BODY>
...the content of this document...
<APPLET CODE="BrowserControl.class" BASE="http://xx.ntt.co.jp" MAYSCRIPT>
<PARAM NAME="1" VALUE="eval:change('RU'),あかあげて,あかあげる">
<PARAM NAME="2" VALUE="eval:change('RD'),あかさげて,あかさげる">
<PARAM NAME="3" VALUE="eval:change('WU'),しろあげて,しろあげる">
<PARAM NAME="4" VALUE="eval:change('WD'),しろさげて,しろさげる">
</APPLET>
</BODY>
<HTML>
```

以上の様にして、音声に反応してJavaScriptが実行される。図2にブラウザ制御アプレットの呼び出しの記述例を示す。アプレットのパラメータには、記述例の様な形で認識単語とJavaScript関数を記述する。各パラメータは数字の名前を持ち[2]、実行すべきJavaScriptの関数、認識単語の読み仮名をカンマで区切って並べたリストを値とする。読み仮名は複数指定して良い。このように、実際にはJavaScriptの関数を呼び出す機能を提供しているだけであるが、関数の定義は文書中で自由に行えるので、実質的に任意のJavaScriptコードを実行できる。図3に上記の記述例のWebページのイメージを示す。このページは、利用者の声の指示に従ってキャラクターが旗を上げ下げするページである。

図3　音声制御可能なWebページ



4　音声認識機能追加プロキシ

音声制御部の機能だけでは、あらかじめブラウザ制御アプレットの呼び出しの記述を含ませたHTML文書に対してしか音声による制御を行うことができない。そこで、HTML文書を解析して自動的にブラウザ制御アプレットの呼び出し記述を追加するプロキシを作成した。このプロキシによって追加される呼び出し記述の例を図4に示す。

図4　プロキシによって追加されるアプレットの呼び出し記述の例

```
<APPLET CODE="BrowserControl.class" CODEBASE="http://xx.yy.zz" MAYSCRIPT>
<PARAM NAME="1" VALUE="link:http://aa.bb.cc/doc1,きたあるぷす,ふゆじたく">
<PARAM NAME="2" VALUE="link:http://aa.bb.cc/doc2,こうつうじこ,ぞうか">
</APPLET>
```

このプロキシは、HTML文書中でリンクの記述が含まれている文を抽出し、形態素解析プログラムによってその文から名詞を抽出して読み仮名を振ることで、図4の様なアプレットの呼び出し記述を生成して、当のHTML文書に追加する。ブラウザがこのプロキシを通してWWWにアクセスすると、自動的にブラウザ制御アプレットの呼び出し記述が追加されるため、任意のWebページに対して音声制御が可能となる。但し、上記のように追加される記述はプロキシ中で設定されたものになるため、一般のWebページに対する音声による指示は既定のものとなる。

上記の仕組みを用いて、さらに単語検索の機能を実現している。この単語検索は、HTML文書中の検索で、「けんさくこうつうじこ」と発声することで、「交通事故」を含む文の位置に表示を切り替える機能である。これは、HTML文書中のNAMEタグを検索し、NAMEタグに続く文章に含まれる単語が発声されると、そのNAMEタグの位置へ表示を切り替える様なアプレット呼び出し記述を生成することで実現している。

---

[2] これは、他のパラメータと区別するためと、パラメータの数が不定であることに対処するためである。

5 課題

本システム音声制御部はほとんどJavaを用いて記述されているが、音声入力の部分と音声認識プロセスとの通信部分についてはマシン依存のコードを用いている。そのため、UNIX上でしか動作しない。また、ブラウザの他に音声入力プロセスを動かさなければならない。しかも、このプロセスとアプレットが通信する必要があるため、セキュリティのためのアクセス制限を回避するために複雑な仕組みが必要となった。さらに、音声認識サーバを用いているため、このサーバに接続可能な範囲でしか用いることができない。これらのことから、現状ではポータビリティに欠けている。ポータビリティのためにはアプレットだけで上記の機能を実現できることが望ましい。その他、プロキシにおける形態素解析の処理速度は約100Kbyte／秒と十分に速いため、インターネットを介したHTML文書の読み出しの速度にはほとんど影響を与えないが、音声認識には数秒を要するため、使い勝手の点ではまだ満足のゆくレベルには達していない。今後、これらの点を解決することが課題となる。

ポータビリティの実現のためには、アプレット以外の部分を全てプラグインに押し込める方法も考えられる。しかし、SUN microsystems社の報道発表[9]によると、Javaの今後の機能拡張の予定の中に音声認識APIが含まれている。この機能を用いれば、上記の課題はほぼ解決できると思われる。そこで我々は、この音声認識APIの実現を待ち、音声制御部をアプレットのみで実現する予定である。

6 おわりに

本論文では、既存のWWWブラウザに音声制御機能を加える方法を提案した。我々のシステムでは、音声に反応するWebページを容易に作成することができる。また、Webページ毎に任意のJavaScriptコードを音声によって起動するように設定しておくことができるため、様々な応用に適用可能である。今後は、ポータビリティを高めたシステムとし、普及を図りたい。

参考文献

[1] http://www.speech.apple.com

[2] http://www.ibm.co.jp/pspjinfo/voice30

[3] http://www.surftalk.com

[4] 土肥, 石塚: "WWW/Mosaicと結合した自然感の高い擬人化エージェントイン タフェース", 電子情報通信学会論文誌, Vol.J79-D-II, No.4, pp.585-591 (1996.4)

[5] H.Dohi, M.Ishizuka: "Visual Software Interface with Realistic Face and Voice-controlled Netscape," Proc. Int'l Conf. on Computational Intelligence and Multimedia Applications (ICCIMA'97), pp.225-229, Gold Coast, Australia, (1997)

[6] 中嶋、加藤: "クリックを使わないマウスの動きと音声を入力とするインタフェース," 情報処理学会研究会報告 SLP, (1996.2)

[7] 渕 武志, 松岡 浩司, 高木 伸一郎: "保守性を考慮した日本語形態素解析システム," 情報処理学会 研究技術報告, SIG-NL97-4, pp. 59-66, (1997.1).

[8] 山田 智一, 野田 喜昭, 井本 貴之, 嵯峨山茂 樹: "クライアント・サーバ構成のHMM-LR連続音声認識システムとその応用," 情報処理学会 研究技術報告, SIG-SLP94-5, pp. 39-46, (1995.2).

[9] http://www.sun.com/smi/Press/sunflash/9612/sunflash.961211.15491.html