

# 音声コマンドを用いた3層アーキテクチャによる遠隔操作システム -第2報 シン・クライアントへの対応と機能サービスの動的な利用-

後藤和弘\*・佐藤辰雄\*\*・大城英裕\*\*\*・吉岡孝\*\*\*\*・本田久平\*\*\*\*\*・築根秀男\*\*\*\*\*

\*情報産業部・\*\*大分県・産業技術総合研究所研究交流センター・\*\*\*大分大学・\*\*\*\*大分県立芸術文化短期大学・  
\*\*\*\*\*大分高専・\*\*\*\*\*産業技術総合研究所

## Remote Control System with Three-Tier Architecture Using Voice Command -Development of Thin-Client and Dynamic Utilization of Distributed Services on Network-

Kazuhiro GOTO\*, Tatsuo SATO\*\*, Hidehiro OHKI\*\*\*, Takashi YOSHIOKA\*\*\*\*, Kyuhei HONDA\*\*\*\*\* and Hideo TSUKUNE\*\*\*\*\*  
Information Technology Division, \*\*Oita-AIST Joint Research Center, \*\*\*Oita University, \*\*\*\*Oita Prefectural College of Arts and Culture,  
\*\*\*\*\*Oita National College of Technology, \*\*\*\*\*AIST

### 要旨

音声コマンドを用いた移動ロボットシステムにおいて、シン・クライアントへ対応するために、クライアントへのソフトウェアの配布、音声データの転送方法、音声データの処理について検討した。また、Webサービスの仕組みをロボットシステムへ適用し、利用者があらかじめネットワーク上の音声認識エンジンの場所や仕様を知らない場合でも、動的に音声認識エンジンを検索・利用できるシステムを開発し、実験によってシステムを評価した。さらに、利用者が屋内を移動する場合やモバイル環境でもシステムを利用できるように、PDAによるクライアントを開発した。

### 1. はじめに

生活空間にはさまざまな機器やシステムがあり、リモコンのボタン・スイッチなどで操作できるものの、操作対象が増えるとともに操作器もまた増加し、使いやすい環境ではなくなってしまう。また、コンピュータの操作にはキーボードやマウスなどを用いるが、誰でもが簡単に使えるインタフェースではない。このような中で、利用者への負担が少なく、自然なコミュニケーションによって操作する手段として、音声によるマンマシンインタフェースが研究されている。利用者への負担を軽減する例としては、ジョイスティックの操作が困難な電動車椅子利用者を支援するために、音声で操作可能な電動車椅子が開発されている<sup>[1]</sup>。一方、ロボットのような複雑なシステムに作業を行わせるために、音声での対話による作業指示が研究されている<sup>[2,3]</sup>。近年注目されている家庭やオフィスにおいて人を支援するロボットも、音声による自然なコミュニケーション機能を持つものが多い。

我々はこれまでに音声データをネットワークで転送し、アプリケーションサーバで音声認識を行うことによって移動ロボットを遠隔操作するシステムを開発してきた<sup>[4]</sup>。データベースシステムなどに採用される3層アーキテクチャをモデルとすることで、システムは拡張性、保守性に優れている<sup>[5]</sup>。このように、ネットワークを利用するシステムでは、他のシステムと柔軟に連携するために標準的な技術を採用することが重要であり、ネットワーク・ロボティクスの分野でも、JavaやCORBA、XML

などを採用したシステムについて研究されている<sup>[6,7,8,9]</sup>。

本研究では、ネットワーク上のどこからでも音声コマンドを用いて移動ロボットや機器等を遠隔操作することを目標とし、Webサービスなどの標準技術を用いて従来のシステムを拡張する。はじめに、Webブラウザをユーザインタフェースとするために、ソフトウェアの配布やデータの転送方法について検討する。また、利用者がネットワーク上の音声認識エンジンを必要に応じて動的に検索・利用できるシステムについて検討する。

### 2. 遠隔操作システム

#### 2.1 システムの概要

遠隔操作システムの概要をFig.1に示す。マイクを接続したクライアント、アプリケーションサーバ、パンチルトカメラを搭載した移動ロボットをネットワークに接続して、システムを構成する。クライアントは、マイクから入力された音声コマンドをアプリケーションサーバ

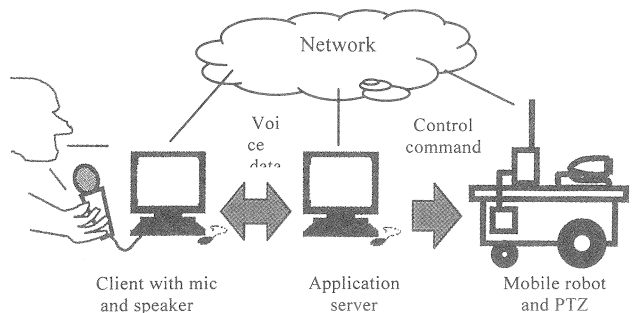


Fig.1 Overview of the system

へ送信する。アプリケーションサーバは受信した音声コマンドをもとに音声認識を行い、対応する制御コマンドをロボットへ送信する。ロボットは受信した制御コマンドにもとづいて、移動やカメラのパンチルト制御を行う。音声認識エンジンには、不特定話者に対応している大語彙連続音声認識デコーダ Julius<sup>[10]</sup>を使用する。本研究では、ロボットを操作するための10種類のコマンドと音声認識結果を比較するようソースを変更して使用する。

## 2.2 シン・クライアントへの対応

シン・クライアントは主にユーザインタフェースに関する処理を行い、通常は Web ブラウザや PDA が用いられる。本システムでは音声の入出力機能とソケット通信機能を実装する。音声を入出力するにはサウンドカードなどハードウェア面で制約されることがあるため、本研究では次の3点について対策をおこなう。

### 2.2.1 クライアント用ソフトウェア

ソフトウェアをネットワーク上から容易にインストールできるように、音声インタフェースを ActiveX コントロールとして開発した。これにより、利用者が Web ブラウザで WWW サーバにアクセスすると、必要なソフトウェアが自動的にダウンロードされ、インストールされる。Fig.2 はシステムの構成と処理フローを表している、図中の数字の(-r)はクライアントへの応答、(-c)は操作指令を

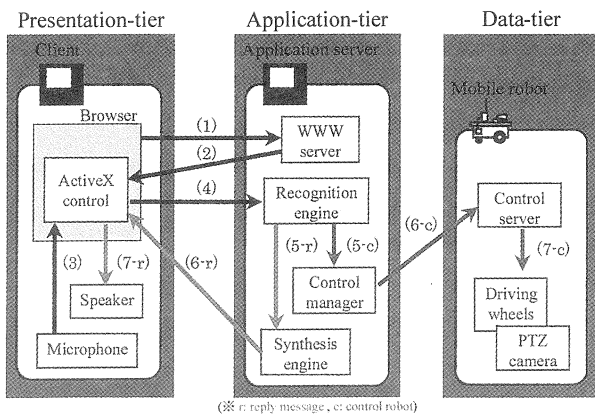


Fig.2 System configuration and data flow

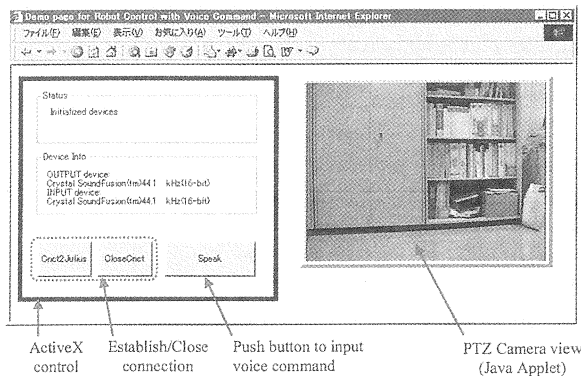


Fig.3 User interface using web browser

それぞれ表す。利用者は Fig.3 に示すように Web ブラウザを使用し、ロボットに取り付けたカメラの映像を見ながら ActiveX コントロールから音声コマンドを入力する。

### 2.2.2 音声データの転送

音声インタフェースでサンプリングした音声データをネットワークで転送する場合、コリジョンによるパケットの欠落は音声認識処理へ影響するため、通信プロトコルには信頼性のある TCP を使用する。クライアントは、はじめにサンプリング周波数や分解能など、WAV ファイルの WAVEFORMATEX 構造体に相当する情報をアプリケーションサーバへ送信する。本システムでは独自に構造体を定義し、アプリケーションサーバからクライアントへの応答形式（音声や文字列など）についてもここで指定する。音声のサンプリングでは、デバイスドライバから WAVE バッファへ音声データが順次格納される。クライアントは、バッファが一杯になった時点でアプリケーションサーバへ音声データを転送する。このとき、Nagle アルゴリズムによる遅延を防ぐため、ソケットオプションに TCP\_NODELAY を指定する。パケットの送出間隔は以下の関係で求められる。

$$t[\text{sec.}] = \frac{8 \times \text{バッファサイズ} [\text{bytes}]}{\text{サンプリング周波数} [\text{Hz}] \times \text{分解能} [\text{bit}]}$$

音声を 16000Hz, 16Bit でサンプリングし、1kbytes のバッファを用意する場合、パケットは約 30ms 毎に送信される。Fig.4 にデータ転送のタイミングを示す。利用者が音声コマンドの発話を終えた時点で、音声データの転送もほぼ終了するため、音声データを一括で送信する場合と比較して全体の時間を短縮できる。低速なネットワーク回線を利用する場合や、組込み機器などクライアントが利用できるメモリが少ない場合には有効である。

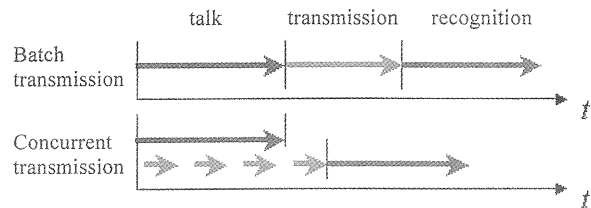


Fig.4 Transmission of voice data

### 2.2.3 アプリケーションサーバにおけるデータ処理

音声認識エンジンへの入力データ形式は、通常、特定のフォーマットだけに対応している。一方、クライアントではサウンドカードなどのハードウェア面からサンプリング周波数などのパラメータが制約されることがあり、音声データをそのまま認識処理に使用できるとは限らない。そこで、アプリケーションサーバでは、サンプリング周波数や分解能などを変更し、音声データを加工

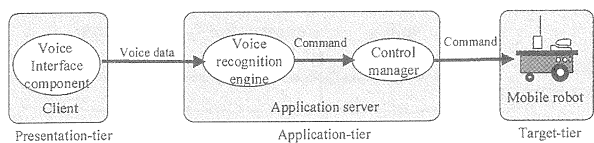
することで、認識エンジンの入力に対応したデータ形式へ変換する。これにより、クライアントはハードウェアが対応している任意の形式でサンプリング可能となる。また、狭帯域の回線を利用する場合にはサンプリング周波数や分解能を低く設定することで、転送する音声データの総量を軽減することができる。

### 2.3 Web サービスでの音声認識エンジンの検索と利用

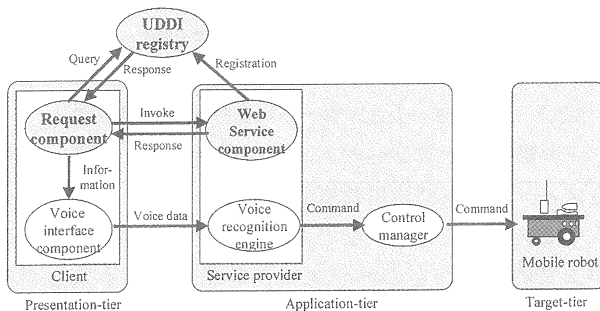
Web サービスは、ネットワーク上のさまざまなサービスをシステムやアプリケーションが利用するための技術である。ソフトウェアシステムのインタフェースやバインディング情報が XML で記述され、ネットワーク上の場所を URI で特定でき<sup>[11]</sup>、SOAP, WSDL, UDDI など XML をベースとする技術を用いている。SOAP (Simple Object Access Protocol)は情報を交換するためのプロトコルで、以下の3つの部分で構成される<sup>[12]</sup>: 1)エンベロップ, 2)エンコーディングルール, 3)リモートプロシージャコールを表現するための規定。WSDL (Web Services Definition Language)は Web サービスを記述する XML 言語で<sup>[13]</sup>、クライアントはこれをもとにリモートプロシージャ名やパラメータなどの情報を取得できる。UDDI (Universal Description, Discovery and Integration)は Web サービスの登録や検索に利用する。UDDI レジストリへ登録したサービスは他のシステムから検索でき、WSDL で記述されたリモートプロシージャは SOAP を用いて実行される。

#### 2.3.1 遠隔操作システムへの適用

これまでに開発してきたシステムの構成を Fig.5(a)に示す。本研究では Web サービスの機能を追加して、Fig.5(b)のようにシステムを拡張する。Web サービスの登録と検索のために UDDI レジストリを追加する。Web サービスに関するソフトウェアコンポーネントと音声認識



(a) The mobile robot system using three-tier architecture



(b) The mobile robot system using web service

Fig.5 Architecture of mobile robot systems

エンジンでサービスプロバイダを構成し、サンプリング周波数や対応言語など、音声認識エンジンの仕様を返すリモートプロシージャを実装する。クライアントへ追加した Web サービスに関するソフトウェアコンポーネントは、UDDI レジストリ、サービスプロバイダと通信を行う。クライアントは音声認識サービスについて UDDI レジストリへ問合せ、その応答をもとに音声認識エンジンの仕様を得るためにサービスプロバイダへリモートプロシージャの実行を要求する。そして、取得した仕様がクライアントのハードウェアでサンプリング可能なフォーマットに対応すれば、音声データを転送し、ロボットの遠隔操作を行うことができる。

Fig.6 に遠隔操作を行う際のシーケンスを示す。図中の 1)~6)は Web サービスの検索, 7)~10)はリモートプロシージャの実行, 11)~14)はロボットの操作をそれぞれ表している。Web サービスは XML 形式のメッセージ交換をおこなうのに対して、音声データはバイナリ形式として転送される。Web サービスでバイナリ形式のデータを転送するには、SOAP Messages with Attachments や DIME (Direct Internet Message Encapsulation) など、幾つかの手法があり、いずれも MIME 構造を利用してバイナリデータを SOAP メッセージへカプセル化する。しかし、SOAP Messages with Attachments はデータのストリーミングには対応していない。また、これらの技術は幾つかのソフトウェアツールキットだけに限定されている。これらの点を考慮し、本システムではバイナリデータを XML 形式にエンコードする際のオーバーヘッドは音声データの転送には向かないと判断し、特定のポートによるソケット通信で音声データを転送する。

#### 2.3.2 クライアント用ソフトウェア

クライアントには、Web サービス、音声インタフェース、ネットワークインタフェースなどの最低限の機能を実装する。音声インタフェースは ActiveX コントロールとして開発しているので、アプリケーションや Web ブラ

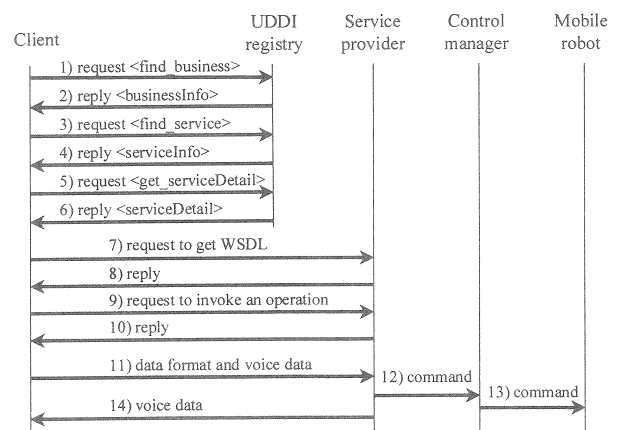


Fig.6 Sequence for remote operation

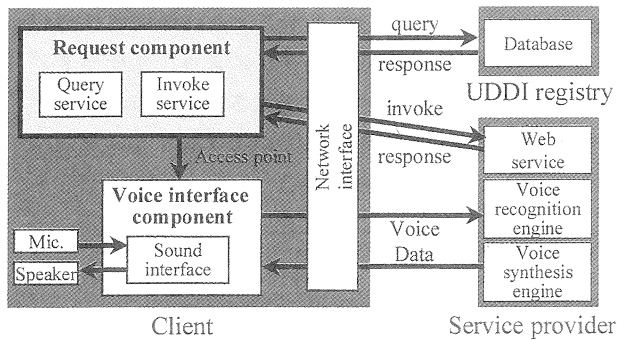


Fig.7 Software components in the client

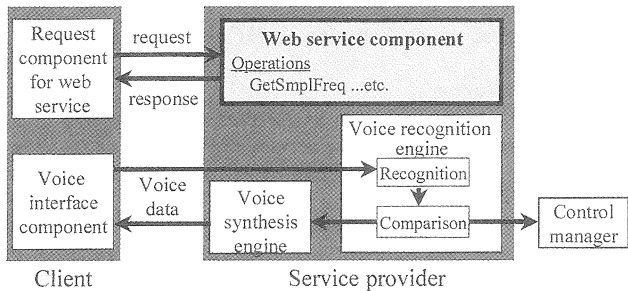


Fig.8 Software components in the service provider

ウザへ容易に埋めこむことができる。Web サービスに関するソフトウェアコンポーネントは、UDDI レジストリやサービスプロバイダとそれぞれ通信を行う。サービスプロバイダから音声認識エンジンのアドレスを取得すると、アドレスを ActiveX コントロールへパラメータとして渡し、音声データの送信先とする。クライアントにおけるソフトウェア構成を Fig.7 に示す。

### 2.3.3 音声認識サービス

Fig.8 に示すように、Web サービスのリモートプロシージャに関するソフトウェアコンポーネント、音声認識エンジン、音声合成エンジンによって、音声認識サービスを提供するためのサービスプロバイダを構成する。Web サービスのリモートプロシージャは、音声認識エンジンの対応するサンプリング周波数、分解能、対応言語、そして、音声データをソケット通信で転送するためのアクセスポイントを戻り値として返す。クライアントは、リモートプロシージャ名や戻り値の変数型などを記述した WSDL の情報をサービスプロバイダから取得し、記述された情報に基づいてリモートプロシージャを実行する。音声合成エンジンは利用者に音声で応答するために使用する。これらのコンポーネントは物理的に異なるコンピュータへも配置できる。

## 3. 実験

### 3.1 概要

実験システムを Fig.9 のように構成し、Web サービスを利用した遠隔操作実験をおこなった。移動ロボットは

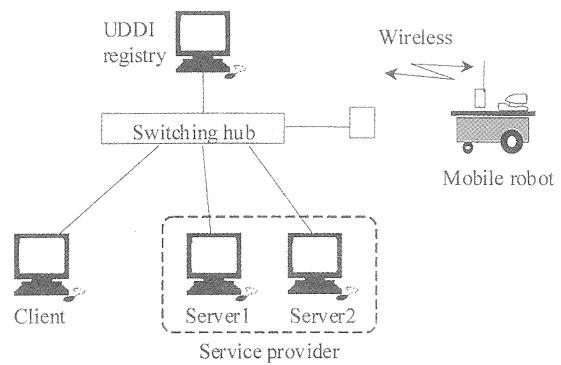


Fig.9 System configuration for experiment

無線 LAN に、他のコンピュータはスイッチングハブにそれぞれ接続する。サービスプロバイダは2台のコンピュータで構成する。Web サービスに関する機能はサーバ1へ実装し、音声認識エンジンはサーバ2へ実装する。クライアント、サービスプロバイダ、UDDI レジストリに使用したコンピュータ、およびソフトウェアツールキットを Table1 に示す。Web サービスを実装するには、Java など幾つかの選択肢があるが、クライアントの ActiveX コントロールとの連携を考慮して.NET 環境を採用した。

実験では、クライアントが UDDI レジストリ、サービスプロバイダと通信を行う際に、送受信するパケットをキャプチャする。また、UDDI レジストリへ問合せを行う際のラウンドトリップタイムを計測する。

Table 1: Software toolkit and specification of the client, the service provider, and the UDDI registry used for the experiment

Client	UDDI registry
Pentium4 (1.3GHz), 384MB Windows2000 professional Microsoft UDDI .NET SDK MS SOAP Toolkit MS VB.net	Pentium3 (1GHz), 512MB Windows2000 server IBM Web Services Toolkit
Service provider	
Server1 (for web service) Pentium3 (1GHz), 512MB Windows2000 server MS IIS5.0 MS VB.net Server2 (for voice recognition) Pentium3 (700MHz), 128MB Linux2.2.12	

### 3.2 音声認識サービスの検索結果

Fig.10 はパケットのキャプチャ結果で、それぞれ HTTP に関する部分と SOAP メッセージに関する部分を含んでいる。Fig.10(a)は、クライアントが日本語に対応した音声認識エンジンを検索するために UDDI レジストリへ送信したパケットである。音声認識サービスを検索するために <find\_service> タグを使用し、サービス名の一部

(voice%Japanese) をメッセージで指定している。

応答として UDDI レジストリからクライアントへ送信されたパケットを Fig.10(b)に示す。SOAP メッセージには、完全なサービス名や一意に割りあてられるサービスキーなどの情報が含まれることが分かる。クライアントはサービスキー、および<get\_serviceDetail>タグを用いて、さらに詳細な情報を UDDI レジストリから取得できる。

```
POST /services/uddi/inquiryapi HTTP/1.1
Content-Type: text/xml; charset=utf-8

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope
<soap:Body>
<find_service generic="1.0"
businessKey="EA63B310-D446-11D6-9248-B80B0335
30A3"
xmlns="urn:uddi-org:api">
<name>voice%Japanese</name>
</find_service>
</soap:Body>
</soap:Envelope>
```

(a) HTTP request including SOAP message that is sent from the client to the UDDI registry

```
HTTP/1.1 200 OK
Content-Type: text/xml

<?xml version="1.0" encoding="UTF-8" ?>
<Envelope
xmlns="http://schemas.xmlsoap.org/soap/envelope/">
<Body>
<serviceList generic="1.0"
xmlns="urn:uddi-org:api"
operator="www.ibm.com/services/uddi"
truncated="false">
<serviceInfos>
<serviceInfo
serviceKey="6EEA49B0-D9BE-11D6-9248-B80B0335
30A3"
businessKey="3B8D7BF0-D9BE-11D6-9248-B80B033
530A3">
<name>Voice recognition service - Japanese</name>
</serviceInfo>
</serviceInfos>
</serviceList>
</Body>
</Envelope>
```

(b) HTTP response including SOAP message that is sent from the UDDI registry to the client

Fig.10 Example of SOAP messages for discovery of the web service by querying to the UDDI registry

### 3.3 リモートプロシージャの実行結果

クライアントがリモートプロシージャの実行を要求したときのパケットのキャプチャ結果を Fig.11 に示す。Fig.11(a)はサンプリング周波数を取得するためにサーバ1へ送信されたパケットである。<GetSmplFreq>タグはサーバ1に実装したリモートプロシージャ名で、WSDLに記述されている。

Fig.11(b)はサーバ1からクライアントへ送信されたパケットで、<GetSmplFreqResult>タグによってリモートプロシージャの実行結果がカプセル化されている。結果から、音声認識エンジンが対応しているサンプリング周波数が16000Hzであることがわかる。

```
POST /WebService5/Service1.asmx HTTP/1.1
Content-Type: text/xml; charset=utf-8
SOAPAction: "http://oiri-sv4/GetSmplFreq"
```

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope>
<soap:Body>
<GetSmplFreq xmlns="http://oiri-sv4/" />
</soap:Body>
</soap:Envelope>
```

(a) HTTP request including SOAP message that is sent from the client to the service provider

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope>
<soap:Body>
<GetSmplFreqResponse xmlns="http://oiri-sv4/">
<GetSmplFreqResult>16000</GetSmplFreqResult>
</GetSmplFreqResponse>
</soap:Body>
</soap:Envelope>
```

(b) HTTP response including SOAP message that is sent from the service provider to the client

Fig.11 Example of SOAP messages for invocation of the remote operation in the service provider

### 3.4 システムの評価

SOAP はシリアライズ化やデシリアライズ化のために JavaRMI などと比較して性能が低いとされているが<sup>[4]</sup>、本システムでは Web サービスは音声認識エンジンの仕様を取得するためだけに利用し、遠隔操作には用いない。このため、シリアライズ化などに要するオーバーヘッドはシステムの利用では大きな問題とはならない。実験において、クライアントがサービスを検索するために UDDI レジストリへ送信する SOAP メッセージはおおよそ 700Bytes 程度であり、検索を 10 回試行した際のラウンドトリップタイムの平均値はおおよそ 128ms であった。特定のサービスを検索するには、SOAP メッセージに埋めこまれるキーワードを適切に設定することが重要である。各サービスに一意に割りあてられるサービスキーを事前を知ることは困難であるため、サービス名を検索に利用すると確認などが容易になると考えられる。ロボットなどのシステムに Web サービスを適用するには、検索キーワードの選択とサービス名の命名規則を十分に検討する必要があることが分かった。

#### 4. システムの拡張

利用者が屋内を移動する場合や、屋外でのモバイル環境にいる場合でもシステムを利用できるように、PDAによるユーザインタフェースを開発した。クライアントのユーザインタフェースを Fig.12 に示す。利用者は無線 LAN やモデムなどによってネットワークに接続した WindowsCE 端末を使用し、移動ロボットに搭載したカメラ映像を見ながら音声で指令を与えることが可能となった。本システムは、ユーザインタフェース、操作対象、各種機能をネットワーク上に分散配置しているため、テレビなどの家電製品を操作対象として容易に追加できる。また、GUI 機能が不要であれば uClinux などの組み込み用 Linux を採用することで、機能や質量の面でより軽量のクライアント端末を実現できる。

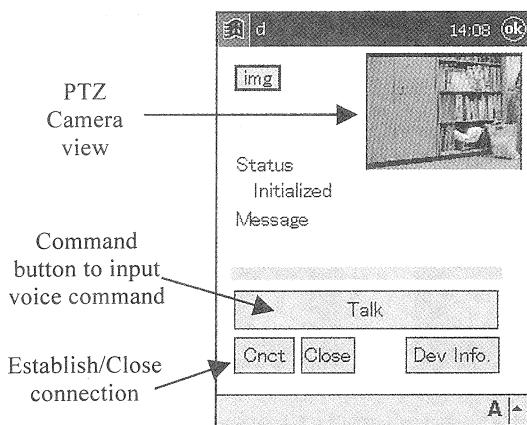


Fig.12 User interface implemented for PDA

#### 5. まとめ

本研究では、ネットワーク上のどこからでも音声による遠隔操作が可能なロボットシステムを実現するために、以下の拡張をおこなった。

- (1) Web ブラウザをユーザインタフェースとし、クライアント用ソフトウェアを ActiveX コントロールとして開発した。
- (2) ネットワーク上に分散配置された音声認識サービスを必要に応じて検索・利用できるように Web サービスの仕組みを適用した。

今回、シン・クライアント、Web サービスへの対応について検討したが、これらは連携しておらず、現在は Web ブラウザからは Web サービスを利用できない。

今後は、これらを結合したシステムを開発するとともに、音声インタフェースを他の ActiveX コントロールと組み合わせ、コンピュータの遠隔操作や Web サイトの閲覧などの応用面についても検討する予定である。

#### 参考文献

- [1] 橋場参生, 中島康博: 音声操作型電動車椅子の開発と評価, 信学技報, WIT2002-66, 2002.
- [2] 村嶋照久, 久野義徳, 島田伸敬, 白井良明: 人間と機械のインタラクションを通じたジェスチャの理解と学習, 日本ロボット学会誌, 18, 4, pp.590-599, 2000.
- [3] 松井俊浩, 麻生英樹, John Fry, 浅野太, 本村陽一, 原功, 栗田多喜夫, 速見悟, 山崎信行: オフィス移動ロボット Jijo-2 の音声対話システム, 日本ロボット学会誌, 18, 2, pp.300-307, 2000.
- [4] 後藤和弘, 佐藤辰雄, 大城英裕, 吉岡孝, 築根秀男: 音声データの転送による移動ロボットの遠隔操作, 第 19 回日本ロボット学会学術講演会講演論文集, 3B33, pp.1163-1164, 2001.
- [5] Kazuhiro GOTO, Tatsuo SATO, Kazuhiro TSURUOKA and Hideo TSUKUNE: Three-Tier Architecture for Remote Control of a Mobile Robot, Proceedings of International Symposium on Robot, pp.159-164, 2001.
- [6] H.Hirukawa and I.Hara: The Web Top Robotics, Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, Robots on the Web Workshop, pp.49-54, 1998.
- [7] S. Jia and K. Takase: A CORBA-based Internet robotic system, Journal of Advanced Robotics, Vol.15, No.6, pp.663-673, 2001.
- [8] 関口大陸, 川上直樹, 柳田康幸, 稲見昌彦, 館暉: RCML: アールキューブ操作言語の開発 (第 10 報) - 多様なクライアント環境への対応 -, 第 19 回日本ロボット学会学術講演会予稿集, 21/22, 2001.
- [9] 大塚隆行, 水川真, 松原安彦, 安藤吉伸: 物理エージェントシステム (PAS) における XML を用いたデバイス接続環境の構築, 日本機械学会ロボティクス・メカトロニクス講演会'02 講演論文集, 2A1-C03, 2002.
- [10] <http://julius.sourceforge.net>
- [11] Web Services Glossary, W3C Working Draft 14 November 2002, <http://www.w3.org/TR/ws-gloss/>
- [12] Simple Object Access Protocol (SOAP) 1.1, W3C Note, May 8, 2000, <http://www.w3.org/TR/SOAP/>
- [13] Web Services Description Language (WSDL) Version 1.2, W3C Working Draft 3 March 2003, <http://www.w3.org/TR/wsd12/>
- [14] M. Govindaraju, A. Slomininski, V. Choppella, R. Bramley, and D. Gannon: Requirements for and Evaluation of RMI Protocols for Scientific Computing, Proceedings of SuperComputing 2000, 2000.